

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»
УДК 004.896

«До захисту допущено»

Завідувач кафедри
_____ І.Р. Пархомей
(підпис)

“ ” _____ 2019 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 126 «Інформаційні системи та технології»

на тему: _____ Робототехнічна система визначення характеристик об'єктів по зображеннях _____

Виконав: студент другого курсу, групи ІК-81мп
(шифр групи)

_____ Гаврилюк Юрій Олексійович _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник доцент, к.т.н., доцент Корнага Я.І. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант НК _____ к.т.н., доцент, Пасько В.П. _____
(назва розділу) (науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент д.т.н., проф., професор каф. ММСА, Мухін В.Є. _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

І.Р. Пархомей

(підпис)

«__» _____ 2019 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Гаврилюку Юрію Олексійовичу
(прізвище, ім'я, по батькові)

1. Тема дисертації «Робототехнічна система визначення характеристик об'єктів по зображеннях», _____

науковий керівник дисертації доцент, к.т.н., доцент Корнага Я.І., _____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «28» 10 2019 р. № 3770-с

2. Термін подання студентом дисертації 18.11.2019

3. Об'єкт дослідження – каскад нейромереж для побудови для побудови каркасних сіток або 3д-моделей об'єктів та алгоритм для оцінки параметрів об'єктів.

4. Предмет дослідження – методи розпізнавання параметрів об'єктів за даними зображеннями.

5. Перелік завдань, які потрібно розробити – аналіз проблеми та існуючих рішень; аналіз і реалізація алгоритму; розробка програмного забезпечення; дослідження ефективності розробленого програмного забезпечення.

6. Орієнтовний перелік ілюстративного матеріалу – шість плакатів

7. Орієнтовний перелік публікацій – одна публікація

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
НК	Пасько В.П., доцент		
Перевірка на співпадіння	Лісовиченко О.І., доцент		

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області	06.09.2019 р.	
2	Постановка задачі	13.09.2019 р.	
3	Аналіз інформаційного забезпечення	17.09.2019 р.	
5	Аналіз алгоритмічного забезпечення	20.09.2019 р.	
6	Розробка алгоритмічного забезпечення	11.10.2019 р.	
7	Розробка програмного забезпечення	01.11.2019 р.	
8	Маркетинговий аналіз стартап-проекту	08.11.2019 р.	
9	Висновки	15.11.2019 р.	

Студент

_____ (підпис)

Ю. О. Гаврилюк

(ініціали, прізвище)

Науковий керівник дисертації

_____ (підпис)

Я. І. Корнага

(ініціали, прізвище)

АНОТАЦІЯ

У роботі розглянуто проблему розпізнавання об'єктів мобільним роботом, оснащеним однією камерою без додаткових датчиків, показано основні особливості існуючих рішень проблеми, їх переваги та недоліки.

Розроблено таку систему, що надає можливість отримання сіткової тривимірної моделі об'єкта та його ключових точок з зображення на основі каскаду двох нейромереж, одна з яких передбачає опорні точки об'єкту у режимі реального часу з використанням згорткової нейронної мережі, а інша оцінює параметри тривимірної сцени на зображенні. Отримана тривимірна модель застосована в методі оцінки лінійних розмірів об'єкту розпізнавання, що реалізований на основі поділу тривимірної моделі на прості компоненти. Дана система може бути використана в робототехнічних системах. Дозволяє зменшити кількість датчиків в роботах.

Ключові слова: тривимірний зір, тримірна реконструкція з зображення, знаходження точки огляду, знаходження опорних точок, сіткова модель, нейромережа.

Розмір пояснювальної записки – 97 аркушів, містить 14 ілюстрацій, 23 таблиці, 2 додатки.

ABSTRACT

Examines the problem of object recognition by a mobile robot equipped with one camera without additional sensors. The basic properties of such a robotic system are determined.

Developed system obtain a 3D object model and its keypoints from an image based on a cascade of two neural networks is considered, one of which provides real-time anchor points of the object using a convolutional neural network and the other estimates the parameters of the 3D scene in the image. The resulting 3D model is applied in the linear object size estimation method, which is implemented by dividing the 3D model into simple components.

Keywords: 3D vision, single image 3D reconstruction, viewpoint estimation, keypoint estimation, wireframe modeling, neural network.

Explanatory note size – 97 pages, contains 14 illustrations, 23 tables, 2 applications.

Пояснювальна записка
до магістерської дисертації

на тему: *Робототехнічна система визначення характеристик
об'єктів по зображеннях*

Київ – 2019 року

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ...	12
1.1 Об’єкт та предмет дослідження.....	12
1.2 Огляд існуючих рішень	13
1.2.1 Meshroom	17
1.2.2 OpenMVG.....	18
1.2.3 Agisoft Metashape	19
1.2.4 RealityCapture	20
1.2.5 PhotoModeler.....	21
1.2.6 3DF Zephyr.....	22
1.2.7 Pix4D	23
1.3 Постановка задачі	25
Висновки по розділу	25
РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ	27
2.1 Нейромережа для генерації каркасів.....	27
2.2 Нейромережа для генерації воксельної моделі 3D-R2N2.....	41
2.3 Алгоритм знаходження параметрів об’єктів.....	49
Висновки по розділу	52
РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	53
3.1 Архітектура програмного забезпечення	53
3.2 Опис структури тестової моделі.....	56
3.3 Вимоги до технічного забезпечення	58

3.4 Вимоги до програмного забезпечення	59
Висновки по розділу	64
РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ	65
4.1 Опис ідеї проекту	65
4.2 Технологічний аудит ідеї проекту.....	67
4.3 Аналіз ринкових можливостей запуску стартап-проекту.....	68
4.4 Розроблення ринкової стратегії проекту	77
4.5 Розроблення маркетингової програми стартап-проекту	80
Висновки по розділу	82
ВИСНОВКИ.....	84
ПЕРЕЛІК ПОСИЛАНЬ	86
ДОДАТКИ.....	88

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

3D – тривимірний

2D – двовимірний

CNN – згорткова нейронна мережа

САПР – система автоматизованого проектування

ГІС – геоінформаційні системи

4D – чотиривимірний

ПЗ – програмне забезпечення

SfM – структура з руху

IoU – індекс перетину

BIM – інформаційне моделювання будівель

DM – деформована моделі

ВСТУП

В сучасному світі технології швидко розвиваються, і в кожному гаджеті є вбудована камера та навіть не дорогі пристрої мають досить хорошу камеру, і технології обробки фотографії можна використовувати в багатьох сферах. Більш того сьогодні будь-який роботизований пристрій може мати камеру при чому інформації з зображення витягалось небагато, тобто роботизованому пристрою потрібні додаткові датчики для роботи з об'єктом або навколишнім середовищем. Саме тому дослідження на тему «Робототехнічна система визначення характеристик об'єктів по зображеннях» допоможе отримати більше інформації з камери для роботизованого пристрою і є актуальною на цей час.

Є різні способи отримати інформації про навколишні об'єкти, але найсучаснішим є використання нейронних мереж, а для обробки зображень – саме згорткових. Протягом останнього десятиліття в області машинного навчання переважають так звані глибокі нейронні мережі, які користуються перевагами підвищення обчислювальної потужності та доступності даних. В роботі розглянуто спосіб отримання тривимірних моделей з зображень, а також їх подальша обробка.

Оцінка 3D-геометрії об'єкта з одного зображення є важливим, але складним завданням у комп'ютерному зорі. Точність визначення точки зору і оцінки форми з 2D-зображень значно поліпшена з використанням машинного навчання, особливо технології «глибокого навчання», наприклад згорткової нейронної мережі (CNN). Однак сучасні методи завжди працюють лише для однієї конкретної категорії об'єктів і показують низьку продуктивність при узагальненні до інших категорій, а це означає, що для випадків зображень об'єктів із декількома класами потрібно кілька детекторів або мереж.

В роботі використано метод із сильною здатністю до узагальнення, який включає лише одну CNN з обробкою DM, що відповідає обробці 3D-точки зору, та оцінці фігур у випадку зображень об'єктів кількох класів. CNN

використовується для виявлення ключових точок потенційного об'єкта на зображенні, тоді як етап узгодження DM, призначений для проведення тривимірного моделювання каркасу об'єкта та оцінки точки зору одночасно з підтримкою виявлених ключових точок. Крім того, оцінка параметрів за допомогою обробки DM має високу відмовостійкість при наявності помилкових ключових точок. Мережі оцінюються на наборі даних Pascal3D+. Експерименти показують, що запропонована мережа добре працює як в точності оцінки параметрів, так і в узагальненні об'єктів для багатьох класів, а алгоритм для оцінки характеристик застосовний до отриманих моделей.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Об'єкт та предмет дослідження

Фотограмметрія – це мистецтво, наука та технологія отримання достовірної інформації про фізичні об'єкти та навколишнє середовище через процес запису, вимірювання та інтерпретації фотографічних зображень та зразків електромагнітних випромінюючих зображень та інших явищ[1].

Існує безліч варіантів фотограмметрії. Один із прикладів – тривимірне вимірювання з двовимірних даних (тобто зображень); наприклад, відстань між двома точками, які лежать на площині, паралельній площині фотографічного зображення, можна визначити, вимірюючи їх відстань на зображенні, якщо масштаб зображення відомий. Іншим є вилучення точних кольорових діапазонів і значень, що представляють такі величини, як альbedo, дзеркальне відображення, металічність або віддзеркалення навколишнього середовища з фотографій матеріалів для фізично обґрунтованого відображення.

Фотограмметрія близької відстані стосується колекції фотографій з меншої відстані, ніж традиційна повітряна (або орбітальна) фотограмметрія. Фотограмметричний аналіз може бути застосований до однієї фотографії, або може використовувати швидкісну фотографію та дистанційне зондування для виявлення, вимірювання та запису складних 2D та 3D-траєкторій руху шляхом подачі вимірювань та аналізу зображень у обчислювальні моделі в спробі послідовної оцінки, збільшуючи точність[1].

Фотограмметрія використовує методи з багатьох дисциплін, включаючи оптику та геометрію проекції. Цифрове захоплення зображення та фотограмметрична обробка включає кілька чітко визначених етапів, які дозволяють генерувати 2D або 3D цифрові моделі об'єкта як кінцевий продукт[2].

Координати 3D визначають розташування точок об'єктів у просторі 3D. Координати зображення визначають розташування зображень об'єктних

точок на плівці або електронному пристрої для зображення. Зовнішня орієнтація [3] камери визначає її розташування в просторі та напрямок її огляду. Внутрішня орієнтація визначає геометричні параметри процесу візуалізації. Це насамперед фокусна відстань лінзи, але може також включати опис спотворень лінзи. Подальші додаткові спостереження відіграють важливу роль: За допомогою шкали, в основному відома відстань двох точок у просторі, або відомі точки фіксації, створюється з'єднання з основними вимірювальними одиницями.

Об'єкт дослідження – каскад нейромереж для побудови для побудови каркасних сіток або 3д-моделей об'єктів та алгоритм для оцінки параметрів об'єктів.

Предмет дослідження – методи розпізнавання параметрів об'єктів за даними зображеннями.

1.2 Огляд існуючих рішень

Також фотограмметрія – це наука про вимірювання фотографій[5]. Вхід до фотограмметрії – це фотографії, а вихідним є, як правило, карта, креслення, вимірювання або 3D-модель якогось реального об'єкта чи сцени. Багато карт, якими ми користуємось сьогодні, створені за допомогою фотограмметрії та фотографій, зроблених із літаків. Фотограмметрію можна класифікувати кількома способами, але одним із стандартних методів є розділення на основі розташування камери під час фотозйомки. Виходячи з цього, ми маємо повітряну фотограмметрію та наземну фотограмметрію. В дисертації нас особливо цікавлять технології для наземної фотограмметрії.

З її допомогою можливо захоплювати великі об'єкти, як-от будівлі або навіть гори, які неможливо було б сканувати іншими методами. Крім того, фотограмметрія також є надзвичайно доступною, оскільки для неї найважливішим обладнанням є камера смартфона. Залишилося лише

програмне забезпечення фотограмметрії для створення сфотографованого 3D-файлу об'єкта[5].

Як і багато інших продуктів, програмне забезпечення для фотограмметрії поставляється в багатьох формах і розмірах. Основні розробники програмного забезпечення опублікували комерційні рішення, ідеальні для промислових та інженерних застосувань. Однак низка програм доступна для безкоштовного завантаження[5].

Як правило, програмне забезпечення фотограмметрії працює таким чином: програма автоматично реєструє спільні точки між кожним зображенням, а потім обчислює відстані між ними в 3D-просторі. Результат – хмара точок, яку можна перетворити в 3D-сітку. Деякі більш досконалі програми також надають інструменти для очищення хмари точок для підвищення загальної якості.

Для того, щоб програмні засоби фотограмметрії працювали належним чином, їм потрібні достатні дані в межах зображень.

«Структура з руху» – це фотограмметрична техніка зображення тривимірних структур із послідовностей двовимірних зображень, які можуть поєднуватися з локальними сигналами руху. Її вивчають у сферах комп'ютерного зору та зорового сприйняття[6]. У біологічному зорі SfM позначає явище, за допомогою якого люди (та інші живі істоти) можуть відновити тривимірну структуру із проєктованого 2D (сітківки) поля руху рухомого об'єкта чи сцени.

Інші програми використовують кодовані маркери. Вони корисні для сканування об'єктів, які мають погані характеристики для обробки або мають відбиваючу та/або прозору поверхню. При правильному застосуванні кодовані маркери можуть забезпечити більш високу точність, ніж «структура з руху».

Більшість програмного забезпечення для фотограмметрії строго розроблені для створення високоякісної 3D-сітки або хмари точок. Однак деякі програми також надають власні інструменти аналізу та роботи, щоб дозволити користувачеві проводити вимірювання, анотувати карту або збагачувати дані.

Такі рішення часто використовуються інфраструктурними компаніями, лісовим господарством, управлінням узбережжя та іншими подібними агенціями.

Зазвичай фотограмметрія передбачає фотоапарат, який керується фотоапаратом або встановлений на штативі для додаткової стійкості зображення. Це називається наземна фотограмметрія, тому що зображення зазвичай знімаються з рівня вулиці. Використання фотограмметрії близької відстані підходить для сканування дрібномасштабних об'єктів. Теоретично великі структури також можна сканувати за допомогою фотограмметрії близького діапазону. Однак часто такі характеристики, як дахи, недоступні з рівня вулиці. У цих випадках доцільно доповнювати фотограмметрію близької відстані повітряною фотограмметрією.

Як випливає з назви, для повітряної фотограмметрії встановлюють камеру на літаку для фотографування великих споруд (будівель, гір, міст) для перетворення цих даних у 3D-модель. Десятиліттями повітряна фотограмметрія була варіантом лише для компаній та установ, які мають ресурси для оренди літака або вертольоту. Повітряна фотограмметрія поступово стає доступною для менших підприємств у багатьох галузях, оскільки ціни на безпілотні апарати знижуються.

Як і у фотограмметрії близької відстані, об'єкт реконструюється пізніше в комп'ютері, і для цього потрібні сотні, якщо не тисячі зображень. Для досягнення якісної тривимірної моделі потрібно, щоб зображення перекривались на 80-90%. Досвідчені пілоти та оператори безпілотних апаратів здатні відповідно регулювати свої маршрути польоту та швидкість. Однак якщо є потреба використовувати дрон, потрібно придбати модель із навігаційним програмним забезпеченням руху по точкам. Ця технологія дозволяє спланувати точний маршрут та швидкість безпілотного апарату перед польотом, щоб забезпечити стабільну якість зображення.

Будівельне проектування і реновація. Коли існуючі будівлі ремонтуються, часто не існує планів, які можна використовувати як орієнтир

для планувальників. У цих випадках програмне забезпечення фотограмметрії є економічно ефективним методом для створення точних планів та вимірювань, які можна імпортувати в програмне забезпечення САПР для початку проекту. Програмне забезпечення для фотограмметрії настільки ж корисне при обстеженні навколишньої території та створенні карт висот.

Оцифровування. Інші захоплюючі галузі, які використовують програмне забезпечення фотограмметрії, – це археологія та збереження культурної спадщини. У минулому археологи вимірювали, а потім простежували будівлі та інші споруди на картах і планах як підняття, фасади та контури. Нещодавно хвиля програмного забезпечення фотограмметрії відкрила нові шляхи запису пам'яток минулих поколінь та дозволила майбутнім поколінням їх вивчати та інтерпретувати.

Геологія та видобуток. Такі дисципліни, як геологія, видобуток та видобуток кар'єрів, також пропонують численні програми для програмного забезпечення фотограмметрії. Порівняно з традиційними методами зйомки, ця технологія є доступним методом для формування хмарних точок та карт висот. Професійні програмні рішення для фотограмметрії надають комплексні аналітичні інструменти для швидкого вимірювання запасів та ям.

Загальна ідея всіх рішень є одна й та сама – отримання тривимірних моделей з фотографій. Використовують для цього, пару способів які відрізняються у обробці фотографій, перший потребує менше сил, і вимагає тільки одної фотографії, але при цьому збільшується час обробки фото, і якість моделі дуже сильно залежить від камери та налаштування ПЗ. На другий спосіб витрачається більше зусиль на фотографії, але так само витрачаються зусилля на обробку фото але при цьому якість буде кращою, проте отриману модель складніше опрацьовувати. Далі приведено існуючі програми які демонструють ці способи на практиці.

1.2.1 Meshroom

Meshroom – це програмне забезпечення з відкритим кодом для фотограмметрії з відкритим кодом, яке створене на базі AliceVision. Програма побудована для простого використання у робочому процесі, який з'єднує всі кроки для створення 3D-моделі. У деяких інших програмах їх потрібно активувати окремо, але оскільки вони утворюють один конвеєр вузлів у Meshroom, вам потрібно лише натиснути одну кнопку («Пуск» у верхній частині робочої області). На рис.1.1 показано повністю інтерфейс програми.

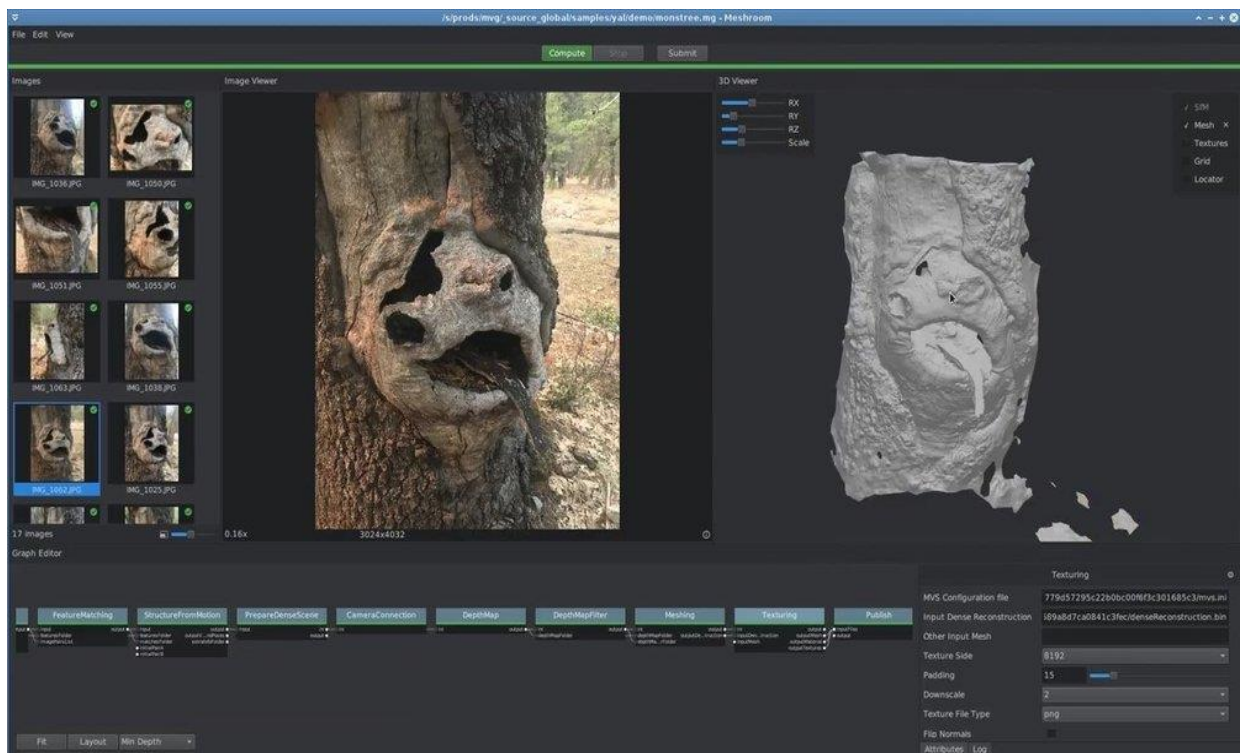


Рисунок 1.1. Приклад інтерфейсу програми Meshroom

Але який сенс мати цей складний конвеєр, коли все, що вам потрібно, – це одна кнопка? Кожен з цих вузлів можна налаштувати, щоб отримати саме той бажаний результат. Наприклад, виберіть вузол текстури, щоб встановити роздільну здатність карти текстури відповідно до програми, від мобільних ігор до візуалізації з високою роздільною здатністю. Чим більше часу вкладено у

вивчення цього безкоштовного програмного забезпечення для фотограмметрії, тим кращими будуть результати.

Користувачі Autodesk Maya можуть використовувати Meshroom безпосередньо у цьому програмному забезпеченні для моделювання 3D як безкоштовний плагін, який також доступний на веб-сайті.

Переваги:

- безкоштовна ліцензія;
- відкритий код;
- можливість інтеграції з САПР системами.

Недоліки:

- необхідність конфігурування;
- потребує серії знімків;
- застарілі алгоритми.

1.2.2 OpenMVG

OpenMVG – це бібліотека для фотограмметрії, яка популярна у спільноті Multiple View Geometry (MVG) та вчених з комп’ютерного зору[7]. Вона зосереджена на технології «структурі з руху» (SfM) фотограмметрії, і має цілий ряд інтегрованих інструментів.

Всі її характеристики та моделі перевірені, щоб переконатися, що вони працюють належним чином у реальних ситуаціях. На рис.1.2 показано повністю інтерфейс програми.

Переваги:

- безкоштовна ліцензія;
- відкритий код;
- популярність та підтримка.

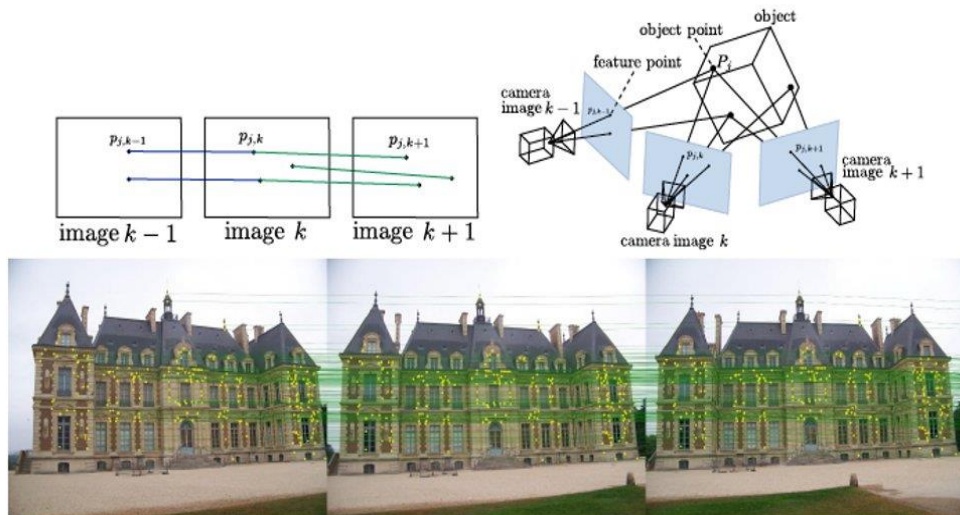


Рисунок 1.2. Приклад інтерфейсу програми OpenMVG

Недоліки:

- необхідність компіляції;
- потребує серії знімків.

1.2.3 Agisoft Metashape

Agisoft Metashape – це професійне програмне забезпечення для фотограмметрії, яке є популярним вибором для застосування в ГІС, документації культурної спадщини, розробки ігор та виробництва візуальних ефектів[8]. Підходить як для ближньої фотограмметрії, так і для повітряної фотограмметрії.

Програма включає комплексні інструменти для редагування хмари точок перед генерацією 3D-сітки. Сюди входить автоматичний класифікатор хмар точок, щоб налаштувати реконструкцію геометрії. Це означає, що програма дозволяє розрізняти різні об'єкти, такі як будівлі та дерева, що дозволяє фільтрувати їх.

Metashape покращує час обробки та якість зйомок у порівнянні з попередніми версіями програми, використовуючи відкриті дані для вдосконалення своїх алгоритмів та машинного навчання. З часом програма

буде продовжувати вдосконалюватися тим більше, чим більше програмою користуватимуться. Крім того, широкі інструменти для вимірювань дозволяють гірничим компаніям, геодезистам та іншим проводити об'ємні вимірювання. Серед інших особливостей є можливість редагування 4D-моделей, а це означає, що цілі сцени можуть бути записані в 3D, а потім змінені[14].

Переваги:

- велика кількість допоміжних інструментів;
- зручність.

Недоліки:

- зайвий функціонал;
- платна ліцензія;
- потребує серії знімків.

1.2.4 RealityCapture

RealityCapture – це фотограмметричне програмне забезпечення, яке створює 3D-моделі з не упорядкованих фотографій (наземних та/або повітряних) або лазерних сканувань без швів[9].

Відмінні можливості, такі як оптимізована обробка, яка сегментує завдання на основі апаратних засобів вашої системи (використовуючи багатоядерну та дискретну графіку), а також безперервний режим чернетки для польових сканувань, інтегроване лазерне сканування та можливість змішування різноманітних медіа даних пропонують комплексне рішення для фотограмметрії, яке вимагає відносно скромного обладнання.

Переваги:

- повноцінне використання апаратних ресурсів комп'ютера;
- зручність.

Недоліки:

- платна ліцензія;

— потребує серії знімків.

1.2.5 PhotoModeler

За допомогою цього інструменту ви можете проводити точні вимірювання та генерувати 3D-моделі для найрізноманітніших цілей[10]. Всередині програмного забезпечення для фотограмметрії ви можете обрати між трьома різними методами для створення вашої моделі: вручну відповідати загальним особливостям між зображеннями, автоматично генерувати 3D-модель з кодованих цілей або генерувати хмари точок із перекриваються зображень. На рис.1.3 показано повністю інтерфейс програми.

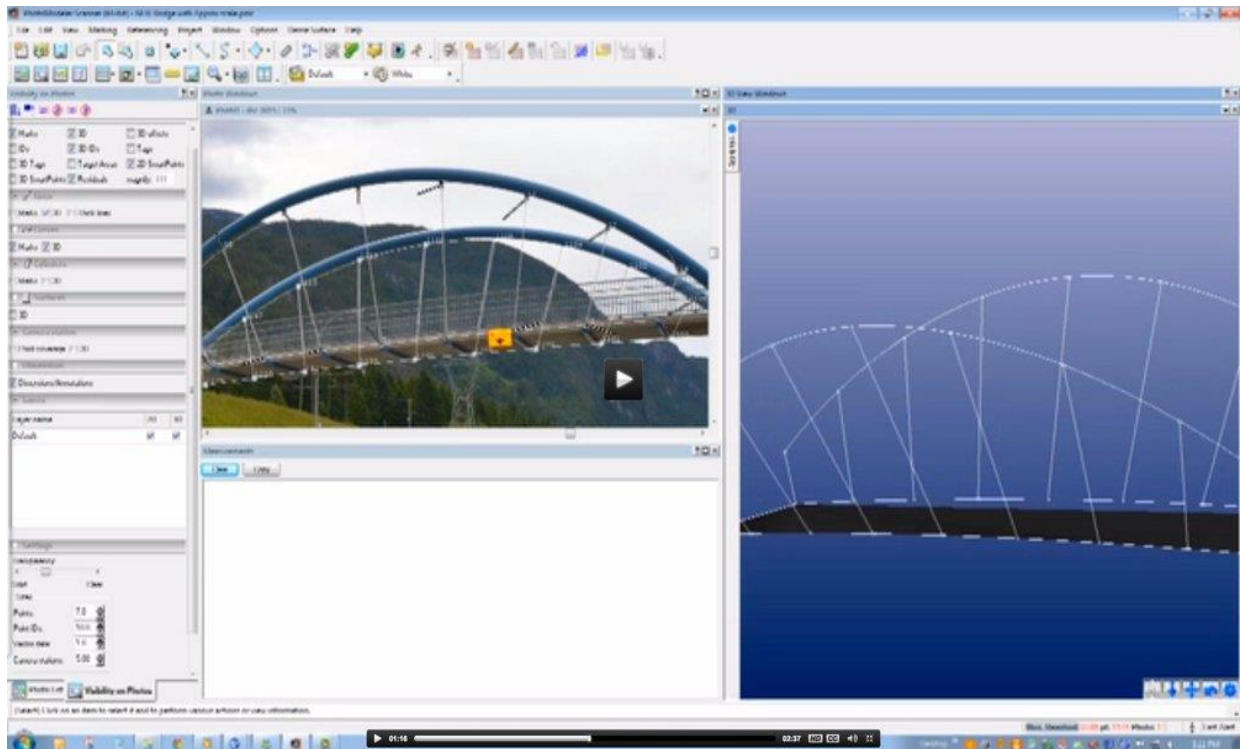


Рисунок 1.3. Приклад інтерфейсу програми RealityCapture

Щоб допомогти вам зрозуміти свої дані, PhotoModeler пропонує цілий спектр аналітичних інструментів. Ви можете вибрати три версії цього програмного забезпечення для фотограмметрії. Стандартна версія, що містить основні функціональні можливості. Версія сканера, яка пропонує

моделювання щільної поверхні (DSM) та відстеження руху на столі. Та версія UAS додає підтримку зображень, знятих безпілотними апаратами.

Переваги:

- генерація моделей у вигляді каркасу, а не хмари точок;
- зручність.

Недоліки:

- платна ліцензія;
- потребує серії знімків.

1.2.6 3DF Zephyr

Це професійне програмне забезпечення для фотограмметрії створене для зручності використання. Її інтерфейс зменшує бар'єри для вступу для початківців та полегшує створення 3D-сканувань за допомогою підказок, що пояснюють процес та допомагають вибрати правильні налаштування. Тим не менш, досвідчені користувачі можуть робити складні налаштування, щоб оптимізувати результат та отримувати вигоду від варіантів експорту в безліч форматів файлів. На рис.1.4 показано повністю інтерфейс програми.

Як і інші передові рішення, 3DF Zephyr здатний узгодити дані фотограмметрії з лазерним скануванням[11]. Таким чином підвищується точність 3D-моделі, зберігаючи також карти текстур. Хоча звичайний процес є повністю автоматичним, можливо втрутитися вручну, якщо потрібно працювати з результатом скануванням у програмі CAD. 3DF Zephyr дозволяє малювати фігури, використовуючи зображення в якості еталону. Отримана САПР модель може бути надіслана в таку програму, як Autodesk Revit для моделювання BIM.

Найкраще, що стосується цього програмного забезпечення для фотограмметрії, це те, що ви можете завантажити безкоштовну версію. Відмінність від платної версії – це максимум 50 зображень, які можна використовувати для реконструкції, менший діапазон інструментів

редагування. Професіонал також повинен ознайомитись із інструментами ГІС та опитування у вдосконалених версіях.

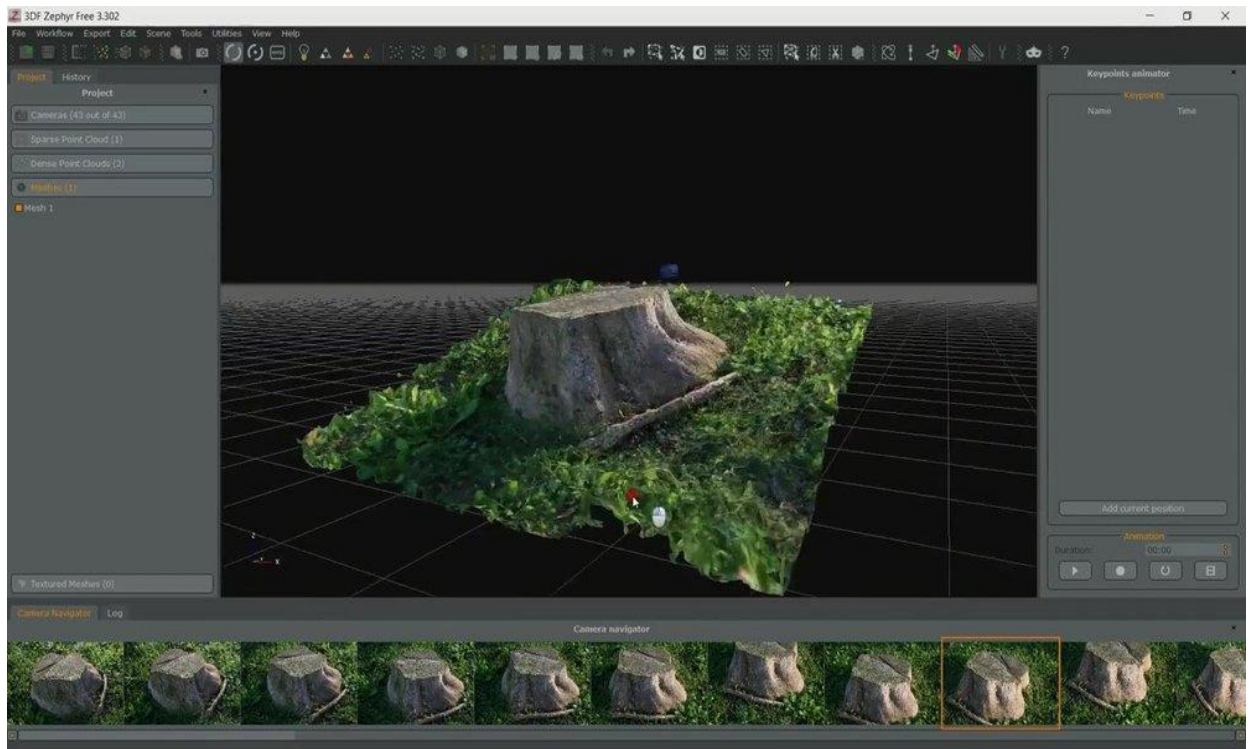


Рисунок 1.4. Приклад інтерфейсу програми 3DF Zephyr

Переваги:

- простота використання;
- підтримка безлічі форматів;
- зручність.

Недоліки:

- платна ліцензія;
- потребує серії знімків.

1.2.7 Pix4D

Pix4D – набір програмних продуктів, які використовують фотограметрію та алгоритми комп'ютерного зору для перетворення DSLR, риб'ячого ока, RGB, теплових та багатоспектральних зображень у 3D-карти та 3D-моделювання[12].

Pix4D – це повноцінне рішення для фотограмметрії. Це означає, що програма не тільки включає в себе генерацію точок хмари, 3D-сітки або карти висот з зображень, але й допомагає в зйомці відповідних зображень. З цією метою Pix4D розробила мобільні додатки для Android та iOS, які перетворюють скромних споживачів дронів на професійні інструменти для картографування. В даний час додатки підтримують дрони DJI, Parrot та Yuneec. Pix4Dcapture контролює шлях польоту безпілота, щоб забезпечити достатнє перекриття між зображеннями для обробки фотограмметрії.

Потім зображення обробляються в програмному забезпеченні Pix4D, яке працює на настільному комп'ютері або в хмарі. Тут ви генеруєте точкові хмари, ортомозаїки, моделі висот, індексні карти тощо. Щоб зрозуміти ваші дані, програмне забезпечення фотограмметрії має комплексні аналітичні інструменти. розробник випустив версії, які задовольняють потреби таких програм, як сільське господарство, геодезичні роботи, архітектура та нерухомість. Наприклад, Pix4Dfields пропонує інструменти для генерування карт донорства та рецептурних карт, які можуть допомогти підвищити вашу продуктивність, оскільки ви можете зосередитись на тих частинах ваших полів, які потребують особливої уваги. Більше того, Pix4D створений для полегшення співпраці. Він підтримує обмін анотаціями, картами та моделями через URL-адреси.

Переваги:

- простота використання;
- наявність мобільного додатку;
- зручність.

Недоліки:

- платна ліцензія;
- зайвий функціонал;
- потребує серії знімків.

1.3 Постановка задачі

Метою роботи є покращення розпізнавання характеристик об'єктів по зображеннях. Для досягнення мети необхідно вирішити наступні задачі:

проаналізувати існуючі методи перетворення зображень в тривимірні моделі та методи оцінки параметрів тривимірної моделі;

вибрати найкращі існуючі методи та модифікувати їх відповідно до задачі;

реалізувати додаток, який розпізнає характеристики об'єктів по їх зображеннях.

Створений додаток повинен відповідати наступним вимогам:

- виділяти об'єкти з зображення;
- створювати тривимірну модель розпізнаного об'єкту;
- виводити на екран отримані з моделі характеристики об'єкту;
- інтуїтивно зрозумілий інтерфейс;
- можливість запуску підсистеми для розпізнавання в роботизованих пристроях;
- простота у використанні;
- простота у налагодженні.

Висновки по розділу

У першому розділі проведено аналіз існуючих рішень для різних видів фотограмметрії. На основі чого були визначені недоліки і переваги цих рішень, а також розглянуто особливості кожного методу. Це професійні пакети програмного забезпечення в якому є багато можливостей, і в них теж є можливість отримати тривимірну модель об'єкта, але для цього потрібно завантажити багато фотографій зроблених з різних ракурсів і рівномірним освітленням. На основі аналізу подібних систем виділено їх основні переваги та недоліки, що і допомогло виявити необхідні функції для подальшої

розробки. Також для роботи цих додатків потрібен ПК, а нам потрібно мати можливість запускати додаток в роботизованому пристрої, що накладає ресурсні обмеження. Сформовано основні вимоги до розроблювальної системи та визначено критерії оптимізації, за якими буде проведене тестування системи.

Виконано постановку задачі системи визначення характеристик об'єктів по зображеннях, в основі якої необхідно розробити аналітичний блок перетворення зображень в тривимірні моделі та оцінка їх лінійних розмірів.

РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Нейромережа для генерації каркасів

У попередні часи комп'ютерного зору окремі предмети, як і цілі сцени, були представлені простими примітивами, такими як багатогранники та узагальнені циліндри. Ці підходи давали багатий опис об'єктів і могли оцінити точки зору та параметри форми, але надійне співставлення їх із захаращеними зображеннями реального світу виявилось важким у той час. З появою комп'ютерів та прогресом у машинному навчанні стало можливим жорстке виявлення предметів та їх частин. В даний час методи, що базуються на зорі, можуть бути широко класифіковані на двовимірні методи, засновані на зображенні та 3D-моделі[16].

Методи розпізнавання зображень використовуються двовимірними методами на основі зображення для спроби безпосередньо відновити інформацію про позу з одного зображення. Ці методи, як правило, працюють за допомогою набору підготовлених представлень моделей, зроблених у діапазоні навколо відомої моделі з різними місцями розташування та точки зору, які завжди страждають від змін у межах категорії. Оцінка пози за допомогою методів, заснованих на 3D-моделі, зазвичай вимагає апріорної 3D-моделі об'єкта, а цілісна вартість витрат визначається, коли модель 3D-деформується підходить до особливостей зображення. Рерік розширив модель деформованої частини (DPM) до 3D, а Xiang представив окремий компонент DPM, відповідний кожній точці огляду. Ці методи можуть оцінювати параметри точки огляду та форми одночасно з різним поданням, включаючи дротяну рамку та 3D-сітку. Останнім часом точність оцінювання та корисність значно покращилися в епоху глибокого навчання. Мережа 3D-інтерпретатора з одним зображенням (3D-INN) представила складну архітектуру конвертної нейронної мережі (CNN) для оцінки 3D-скелета, що містить інформацію з точки зору та форму. Чи Лі отримав 3D-об'єктну структуру завдяки глибокій

архітектурі CNN з знаннями домену у прихованих шарах. Однак ці методи покладаються на анотацію ключових точок, що стосуються конкретної категорії, і не є узагальненою. При роботі з об'єктами, що належать до класу, різні мережі потребують навчання окремо, що ігнорує подібності між категорійної структури та значно збільшує витрати на навчання.

Виявлення ключових точок. Представлення каркасної форми – це стисла форма моделювання структури з сильною здатністю опису, яка може зберегти структурні властивості при 3D-моделюванні. Щоб витягнути каркас із зображення, необхідно виявити ключову точку. Дослідники досягли значного прогресу у виявленні ключових точок. Традиційний спосіб – підготовка класифікатора за допомогою функції ручної роботи, яка використовується в DPM. Останнім часом було кілька спроб застосувати CNN для виявлення ключових точок.

Тошев підготував глибоку нейронну мережу для двовимірної регресії людини. Xiang Yu оптимізував коефіцієнти деформації на основі аналізу основних компонентів (PCA) 2D ключових точок для досягнення найсучасніших показників на обличчі та тілі людини. Незважаючи на хорошу ефективність цих підходів, вони поділяють загальне обмеження: кожна ключова точка підготовлена лише для конкретного типу з конкретного об'єкта. Xingyi Zhou запропонував представлення ключових точок без прив'язки до категорій, яке поєднує багато пікову теплову карту (StarMap) для всіх типів ключових точок у наборі даних Pascal3D+ за допомогою мережі пісочних годинників. Це представлення забезпечує гнучкість представлення різної кількості ключових точок у різних категоріях. Незважаючи на високу ефективність узагальнення, цей метод не може надати семантичну інформацію ключових точок.

Фреймворк для запропонованого способу показаний на рис. 2.1, який складається з двох частин: виявлення ключових точок та обробки узгодженої моделі, що деформується.

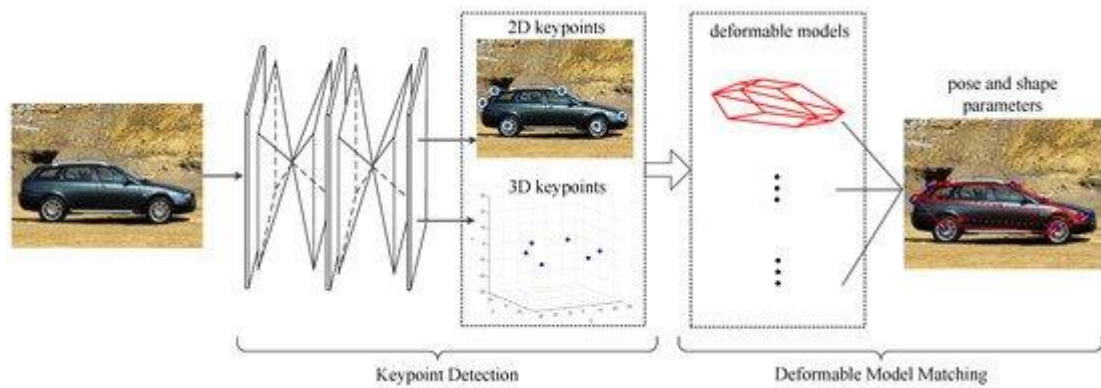


Рисунок 2.1. Ілюстрація фреймворку. Для вхідного зображення через мережу схожу на пісочний годинник отримують 2D ключові точки та їх тривимірні координати. Потім ці ключові точки узгоджуються з деформованими моделями. Після цього виходять параметри точки огляду та форми.

Для виявлення ключових точок найбільш широко використовуваним способом є представлення ключових точок як багатоканальних теплових карт, які пов'язують кожну ключову точку з одним каналом у конкретній категорії об'єктів. Для цих методів, хоча кожен ключовий пункт має семантичний зміст, вони обмежені конкретною категорією. Іншими словами, ключові точки від різних об'єктів повністю відокремлені. Ми прагнемо виявити ключові точки в різних категоріях, тому можна отримати узагальнену мережу для об'єктів, що належать до багатьох класів. Цей підхід натхненний мережевим підходом до виявлення ключових точок, запропонованим Xingyi Zhou. Цей метод дозволяє знаходити всі ключові точки в різних категоріях, використовуючи лише одну мережу, однак отримані ключові точки не мають семантичного значення. Для оцінки тривимірної точки зору потрібно смислове значення кожної ключової точки, щоб відповідати апріорній моделі. В результаті потрібні ключові точки, їх 3D-розташування та глибина, щоб отримати смислове значення ключових точок у роботі Zhou. Мережа в нашому методі схожа на Zhou, тоді як нам потрібні лише 2D ключові точки та їх 3D-розташування під час тренінгу. Це тому, що смислове значення кожної ключової точки можна надати, зіставивши

з деформованою моделлю позаду. Використовувана мережа показана на рис 2.2.

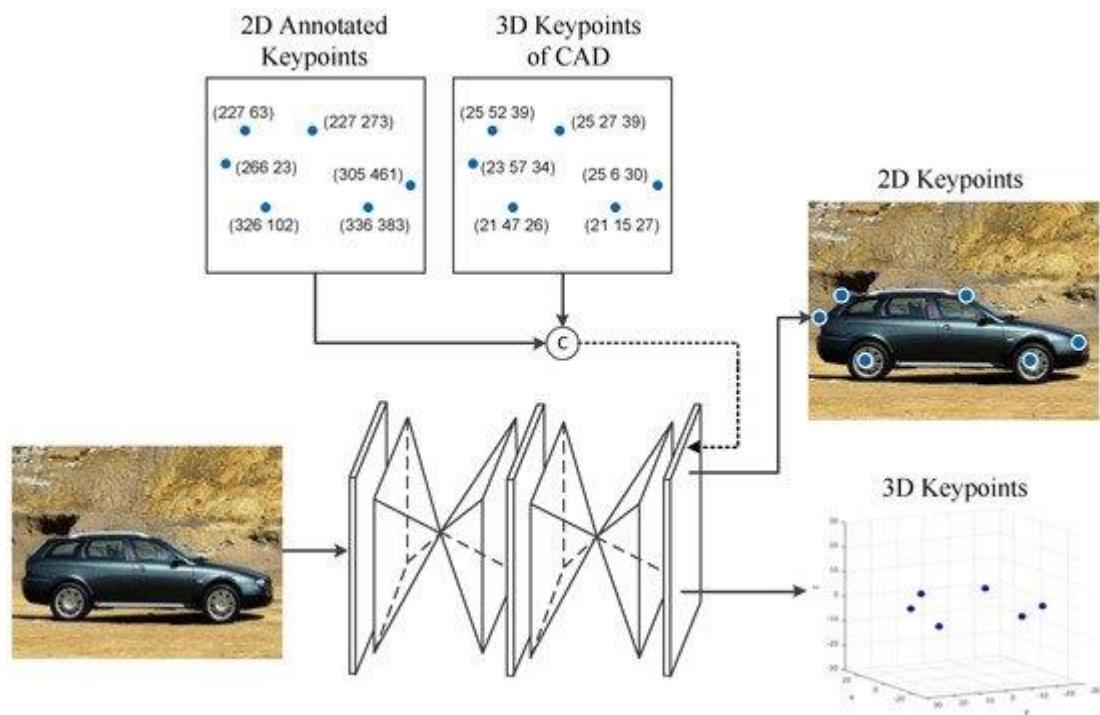


Рисунок 2.2. Мережа пісочного годинника для виявлення ключових точок.

Мережа нашого методу прогнозує 2D ключові точки та їх тривимірні координати.

Дизайн пісочного годинника мотивований необхідністю збору інформації в кожному масштабі. Хоча місцеві докази є важливими для виявлення таких рис, як обличчя та руки, остаточна оцінка пози вимагає цілісного розуміння всього тіла.

Орієнтація людини, розташування кінцівок та відносини сусідніх суглобів є одними з багатьох підказів, які найкраще розпізнаються на різних масштабах зображення. Пісочний годинник – це проста, мінімальна конструкція, яка здатна охопити всі ці функції та об'єднати їх для отримання піксельних прогнозів.

Мережа повинна мати певний механізм для ефективної обробки та консолідації функцій у різних масштабах. Деякі підходи вирішують це за

допомогою використання окремих конвеєрів, які обробляють зображення незалежно з декількома роздільними здатностями та поєднують функції в подальшому в мережі. Натомість ми вирішили використовувати один конвеєр із пропущеними шарами для збереження просторової інформації при кожній роздільній здатності.

Мережа досягає найнижчої роздільної здатності при 4x4 пікселях, що дозволяє застосовувати менші просторові фільтри, які порівнюють функції у всьому просторі зображення. Пісочний годинник встановлюється наступним чином: згортання та максимум шарів об'єднання використовуються для обробки функцій до дуже низької роздільної здатності. На кожному кроці максимального об'єднання мережа відключається і застосовує більше згортків у вихідній попередньо об'єднаній роздільній здатності. Досягнувши найнижчої роздільної здатності, мережа починає послідовність перегляду та комбінації функцій у різних масштабах. Щоб об'єднати інформацію в двох суміжних резольюціях, ми дотримуємося описаного процесу і зробить найближчий сусідський підсилювач нижчої роздільної здатності з подальшим елементарним додаванням двох наборів функцій. Топологія пісочного годинника симетрична, тому для кожного шару, присутнього на дорозі вниз, є відповідний шар, що йде вгору.

Після досягнення вихідної роздільної здатності мережі застосовуються два послідовних раунди поворотів 1x1 для створення остаточних прогнозів мережі. Вихід з мережі – це набір теплових карт, де для заданої теплової карти мережа передбачає ймовірність присутності суглоба на кожному пікселі.

При збереженні загальної форми пісочного годинника все ж є певна гнучкість у конкретному виконанні шарів. Різні варіанти можуть мати помірний вплив на кінцеву продуктивність та навчання мережі. Ми вивчаємо кілька варіантів дизайну шарів у нашій мережі. Недавня робота показала значення кроків скорочення з 1x1 згортками, а також переваги використання послідовних менших фільтрів для захоплення більшого просторового контексту. Наприклад, можна замінити фільтр 5x5 двома окремими фільтрами

3x3. Ми протестували наш загальний дизайн мережі, замінюючи модулі різних рівнів на основі цих розумінь. У нас спостерігалось підвищення продуктивності мережі після переходу зі стандартних згорткових шарів з великими фільтрами і без кроків зменшення до нових методів, таких як залишкові модулі навчання, та конструкції, засновані на «створенні».

Після первинного підвищення продуктивності цих типів конструкцій, різні додаткові дослідження та модифікації шарів мало зробили для подальшого підвищення продуктивності або часу на навчання.

Остаточний дизайн широко використовує залишкові модулі. Фільтри більше 3x3 ніколи не використовуються, а вузьке обмеження обмежує загальну кількість параметрів на кожному шарі, зменшуючи загальне використання пам'яті. Щоб поставити це в контекст повноцінного дизайну мережі, кожне поле являє собою один залишковий модуль.

Операція з повною входною роздільною здатністю 256x256 вимагає значного обсягу пам'яті GPU, тому найвища роздільна здатність пісочного годинника (і, таким чином, кінцева роздільна здатність) становить 64x64. Це не впливає на здатність мережі виробляти точні спільні прогнози. Повна мережа починається з згорткового шару 7x7 з кроком 2, після чого залишковий модуль і раунд максимального об'єднання, щоб зменшити роздільну здатність з 256 до 64. Два наступні залишкові модулі передують пісочному годиннику. По всій пісочній годині всі залишкові модулі видають 256 функцій.

Продовжуємо нашу мережеву архітектуру далі, укладаючи кілька кінцевих годин в кінцевий час, подаючи вихід одного як вхід в наступний. Це забезпечує мережу механізмом для повторних висновків знизу вгору, зверху вниз, що дозволяє переоцінювати початкові оцінки та функції в усьому зображенні. Ключовим у цьому підході є прогнозування проміжних теплових карт, за якими ми можемо застосувати втрати. Прогнози генеруються після проходження кожного пісочного годинника, де мережа мала можливість обробляти функції як в локальному, так і в глобальному контексті. Подальші модулі пісочного годинника дозволяють повторно обробляти ці функції

високого рівня для подальшої оцінки та оцінки просторових відносин вищого порядку. Це аналогічно іншим методам оцінювання пози, які продемонстрували високу ефективність з багаторазовими ітераційними етапами та проміжним наглядом.

Розглянемо межі застосування проміжного нагляду лише з використанням одного модуля пісочного годинника. Яке було б відповідне місце в трубопроводі для створення початкового набору прогнозів? Більшість функцій вищого порядку є лише у нижчих роздільних здатностях, за винятком самого кінця, коли відбувається перебіг вибірок. Якщо нагляд забезпечується після того, як мережа здійснить перемолочування, тоді немає можливості переоцінити ці функції відносно один одного у більш широкому глобальному контексті. Якщо ми хочемо, щоб мережа найкраще уточнювала прогнози, ці прогнози не можуть бути оцінені виключно в локальному масштабі. Відносини до інших спільних прогнозів, а також загальний контекст та розуміння повного іміджу є вирішальним. Застосування нагляду раніше в конвеєрі перед об'єднанням – це можливість, але на цьому етапі особливості даного пікселя є результатом обробки відносно локального сприйнятливого поля і, таким чином, не знають критичних глобальних сигналів. Повторне висновок знизу вгору, згори вниз зі складеними пісочними годинниками зменшує ці проблеми. Локальні та глобальні сигнали інтегровані в кожен модуль пісочного годинника, і для того, щоб попросити мережу зробити раннє передбачення вимагає, щоб вона мала високе розуміння зображення, а лише проходила повну мережу. Подальші етапи обробки знизу вгору та згори вниз дозволяють глибше переглянути ці особливості.

Такий підхід для повернення вперед і назад між масштабами є особливо важливим, оскільки збереження просторового розташування ознак є важливим для виконання остаточного кроку локалізації. Точне положення стику – неодмінний підказ для інших рішень, які приймає мережа. При такій структурованій проблемі, як оцінка позиції, результат є взаємодією безлічі різних особливостей, які повинні поєднуватися, щоб сформувати цілісне

розуміння сцени. Суперечливі докази та анатомічна неможливість – це великі відомості про те, що десь уздовж лінії була допущена помилка, і, переходячи вперед і назад, мережа може підтримувати точну локальну інформацію під час розгляду, а потім переглядати загальну узгодженість ознак.

Ми реінтегруємо проміжні прогнози назад у простір функцій, відображаючи їх на більшу кількість каналів з додатковою згорткою 1×1 . Вони додаються до проміжних ознак пісочного годинника разом із характеристиками, що виводяться з попереднього етапу пісочного годинника. Отриманий вихід безпосередньо служить входом для наступного модуля пісочного годинника, який генерує інший набір прогнозів. У остаточному дизайні мережі використовується вісім стиснень. Важливо зазначити, що ваги не розподіляються між модулями пісочного годинника, а функція втрат застосовуються до прогнозів усіх пісочних годинників.

Навчання мережі за нашим методом вимагає анотацій 2D ключових точок та відповідних їм 3D-місць. Анотації 2D ключових точок на зображення широко доступні у багатьох наборах даних. Анотування 3D-точок ключових точок CAD-моделі також не є важкою роботою з інтерактивним 3D-інтерфейсом, що було зроблено в деяких наборах даних, таких як Pascal3D + і ObjectNet3D набір даних. Порівняно з роботою Zhou, підготовка даних для мережі є більш доцільною. Використовується мережа пісочного годинника з 2 стеками. 2D ключові точки та їх 3D-розташування розподіляються чотириканальними тепловими картами. Під час тренінгу відстань L2 мінімізується між вихідними чотириканальними тепловими картами та їх основною правдою.

Після мережі виявлення ключових точок наступним кроком буде оцінка параметрів об'єкта з ключових точок. Ключові точки, виявлені в мережі, не мають семантичного значення. Таким чином, лише ключових точок недостатньо для оцінки параметрів пози і форми для будь-якого об'єкта. Крім того, для оцінки точки зору та тривимірного моделювання потрібні такі параметри: категорія об'єкта, семантичне значення ключової точки, параметри

форми та тривимірне представлення. Складно отримати всі параметри одночасно.

Для вирішення цих проблем ми пропонуємо метод узгодження моделі, що деформується, показаний на рис 2.3.

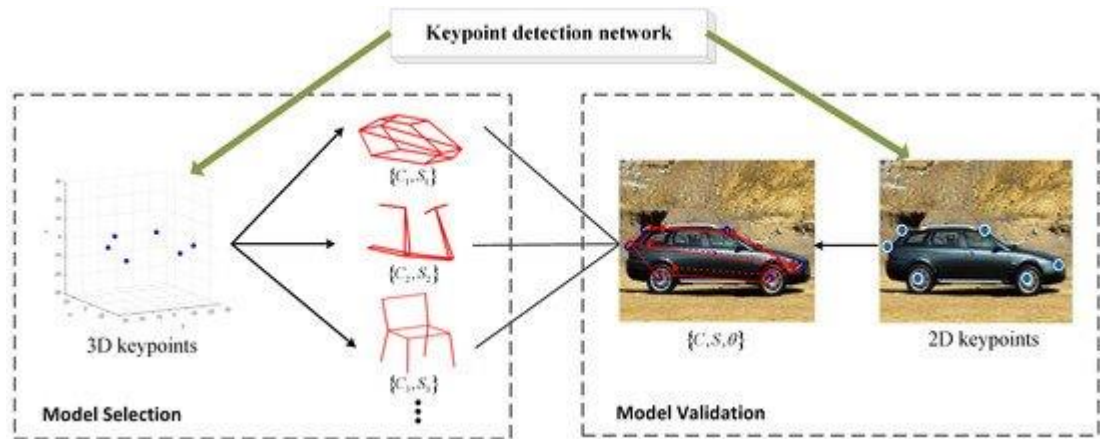


Рисунок 2.3. Обробка сумісної моделі, що деформується. Деформоване узгодження моделі можна розділити на два етапи: вибір моделі та перевірка моделі. Параметри об'єкта отримані з найкращої моделі, що деформується.

Узгодження деформованої моделі можна розділити на два етапи: вибір моделі та перевірка моделі. На першому етапі витягнуті тривимірні ключові точки використовуються для відповідності різним категоріям деформованих моделей для вибору кандидатів деформованих моделей. На другому етапі ці кандидатські моделі перевіряються шляхом узгодження з вилученими 2D ключовими точками, а параметри форми та точки зору можна отримати шляхом оптимізації. Кінцеві параметри об'єкта отримані з найкращої моделі, що деформується.

Формулювання, побудова та оптимізація моделі, що деформується, описані в наступних розділах.

У цьому розділі проілюстровано побудову багатокласових деформуючих моделей. Очікується, що модель, що деформується, матиме можливість захоплювати дисперсію в класі. Ми моделюємо кожну категорію

об'єктів як деформується 3D-каркас, який є стислим та виразним. Під час навчання 3D координати ключових точок, ановані в CAD-моделі, представлені у вигляді вектору, і ми виконуємо PCA на цих векторах для бібліотеки моделей CAD певної категорії. Представлення геометрії базується на середньому дротяному кадрі μ плюс лінійної комбінації r основних компонентів p_k з параметрами геометрії s , де s_k - вага k -го головного компонента:

$$\mathbf{M}(S) = \mu + \sum_{k=1}^r s_k p_k \quad (2.1)$$

3D каркас може визначатися параметром форми S :

$$S = \{s_k\}_{k=1 \dots r} \quad (2.2)$$

r у цій роботі встановлено як 3. Приклад моделі 3D каркаса зображений на рисунку 2.4.

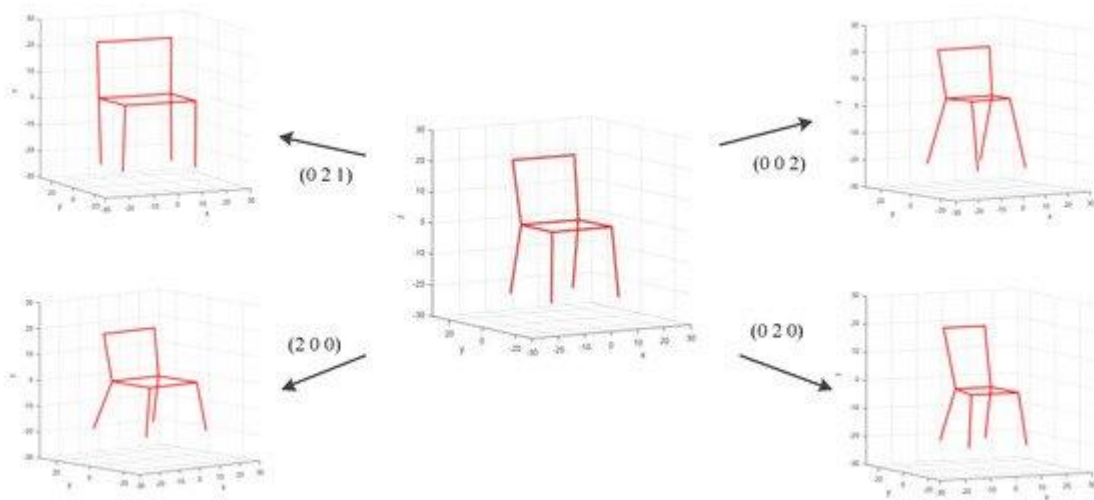


Рисунок 2.4. Деформоване зображення 3D-каркаса. Моделі стільців різних параметрів форми генеруються PCA. Цифри, зазначені на рисунку, – це вагові параметри перших трьох основних компонентних напрямків, які представлені як S .

Для вирішення задач оптимізації в рівнянні (2.1) ми використовуємо алгоритм градієнтного спуску. Розподіл ймовірностей рівняння (2.1) складається з двох відносно незалежних елементів. Щоб оптимізувати результати та вийти з лову локального мінімуму, поетапна оптимізація – це наша стратегія. Для першого пункту ймовірність рівняння отримується шляхом зіставлення 3D-ключових точок з кожною деформованою моделлю. Якісні результати показані на рис 2.5. З результату ми бачимо, що ключові точки добре поєднуються з деформованою моделлю шляхом оптимізації.

Оптимізація другого елемента в рівнянні (2.1) чутлива до початкового значення параметрів точки огляду. Запропонований метод визначення початкового значення за видимими ключовими точками. Взявши приклад автомобіля, рис 2.6 ілюструє детальну процедуру.

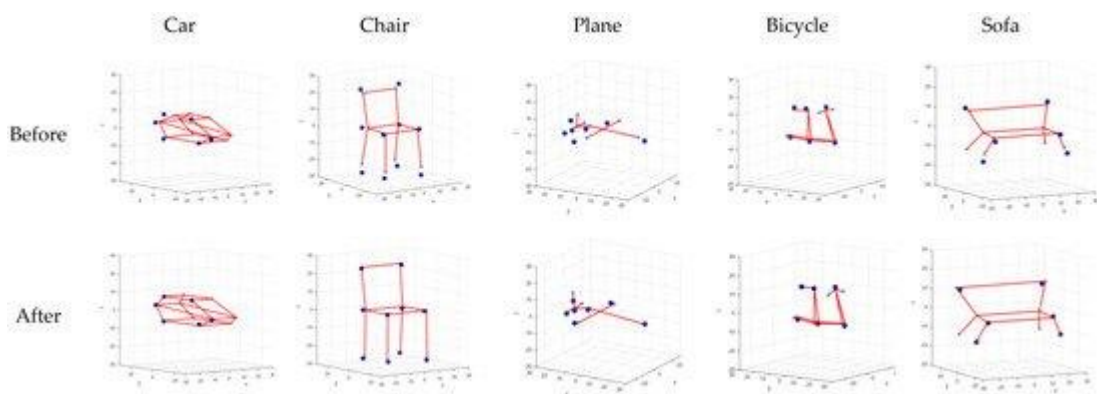


Рисунок 2.5. Якісні результати узгодження 3D-ключових точок з деформованою моделлю. Видно, що 3D-ключові точки та відповідні моделі, що деформуються, добре підходять після відповідності.

Механізм дискримінації необхідний для об'єктів, які мають іншу категорію, ніж апріорні моделі. Для об'єктів, категорія яких не входить до наших апріорних моделей, відхилення проекції деформованої моделі та 2D ключових точок було б більше порівняно з об'єктами, які мають ту саму категорію з відповідною деформованою моделлю. Отже, встановлюється поріг відхилення, щоб оцінити, чи належить він до категорій, що містяться.

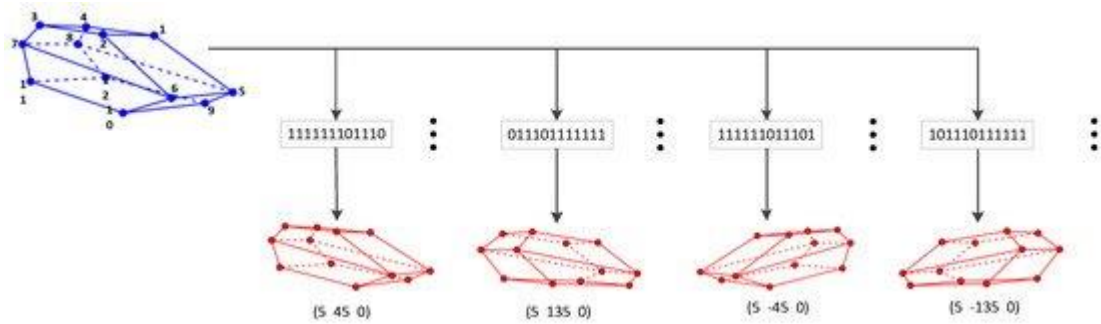


Рисунок 2.6. Метод визначення початкової точки зору. Видимість ключових точок у різних точках огляду зараховується як словник. Для набору точок введення вихідну точку огляду можна отримати, зіставивши їх видимість зі словником.

Короткий огляд методу показано на рис. 2.7. Спочатку ключові точки виявляються із зображення. Потім 3D-ключові точки узгоджуються з кожною моделлю, що деформується, а 2D-ключові точки узгоджуються з проекцією деформованої моделі. Після процесу узгодження оптимальні параметри пози і форми отримують з найкраще підходить деформованої моделі. Наступний крок – визначити, чи належить цільовий тип до існуючих моделей відповідно до відхилення відповідності. Нарешті, отримуються параметри об'єкта.

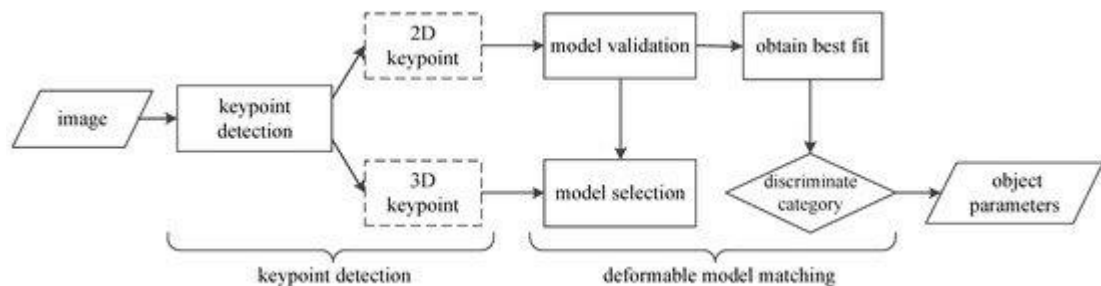


Рисунок 2.7. Підсумок методу.

Ми оцінюємо метод, запропонований на наборі даних Pascal3D+. У цьому розділі ми оцінюємо підхід з двох аспектів: моделювання каркасних

кадрів та оцінка точки зору. Важливо зазначити, що ці два завдання виконуються одночасно в нашому методі.

Моделювання каркасних рам. Чотири стовпчики на рис 2.8. – це вхідне зображення, виявлені 2D ключові точки, відповідність між 3D-ключовими точками та деформованою моделлю та проекція каркасної рамки. У четвертому стовпчику червона лінія являє собою проекцію каркасної рамки, а її відхилення від виявлених 2D ключових точок відображається у вигляді жовтої лінії.

3D-каркасні моделі формуються після узгодження моделі, що деформується. Щоб перевірити надійність, ми штучно вставляємо помилкові ключові точки у отримані мережею. Оцінюється відхилення між проекцією моделі каркасної рамки та основним значенням 2D ключових точок. Візьмемо корпус дивана, результати зображено на рис 2.9.

Ми штучно додавали ключові точки помилок для перевірки відмовостійкості, як показано в першому та другому стовпцях. В (a) немає ключової точки помилки. (b) та (c) показують випадки із доданими ключовими точками помилок. Через відповідний процес, представлений у другому стовпчику, запропонований метод може ідентифікувати помилкові точки, показані як третій стовпчик. Для запропонованого методу відхилення між проекцією моделі каркасної рамки та істинністю основної точки 2D ключових точок не змінюється зі збільшенням помилкових точок, відображених у четвертому стовпці. Останні два стовпці порівнюють відмовостійкість методу та запропонованого методу оцінювання точки зору, який використовує критерії оцінювання в рівнянні.

Видно, що запропонований метод добре обробляє неправильні ключові точки. Через те, що деформується відповідність моделі, результати помилок у мережі, схоже, не викликають значних негативних наслідків.

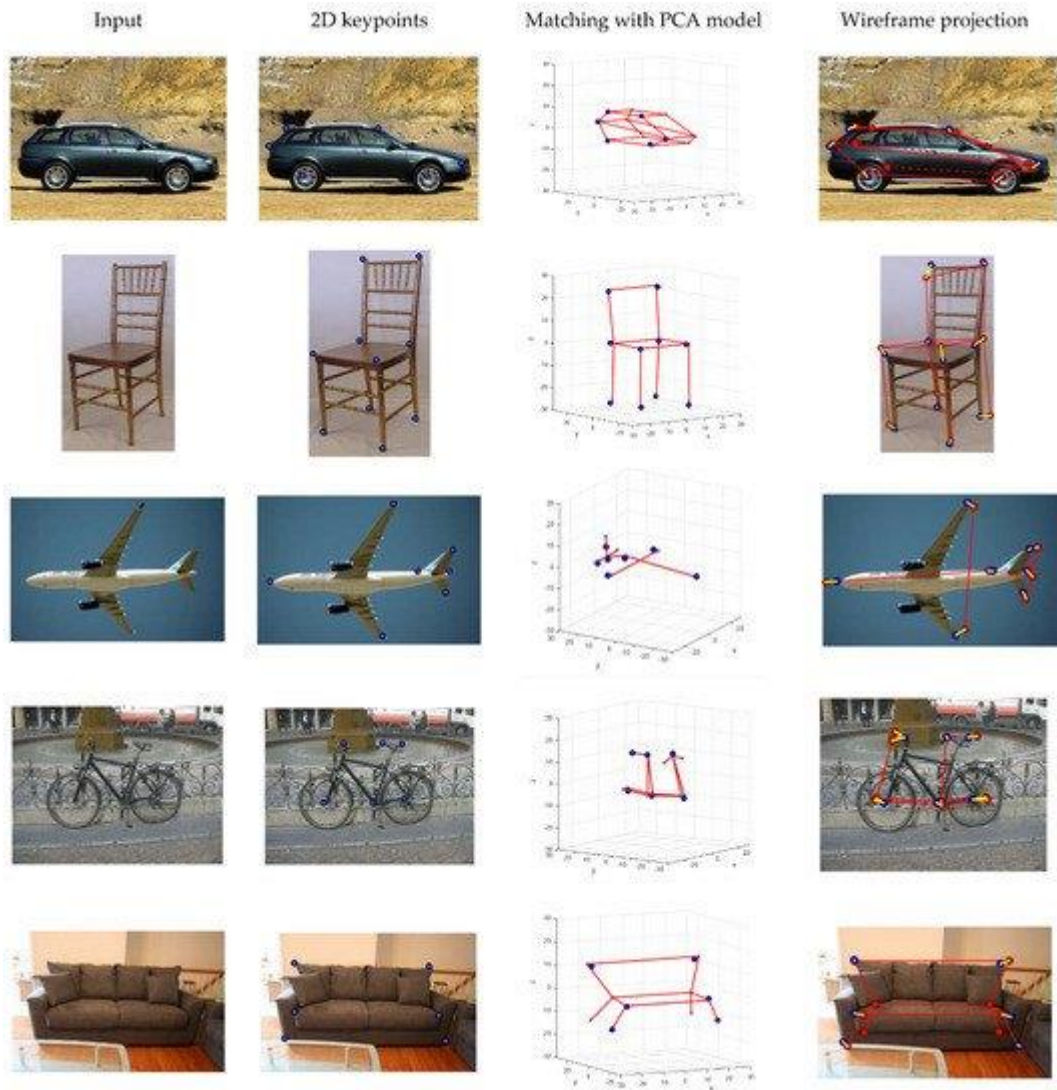


Рисунок 2.8. Якісні результати тривимірного моделювання каркасних рам.


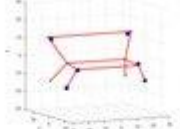





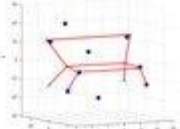

	2D keypoints	Matching with PCA model	Wireframe projection	Deviation	Viewpoint Error	Error of [11]
(a)				39 pixels	8.1	8.3
(b)				39 pixels	8.1	8.3
(c)				39 pixels	8.1	10.8

Рисунок 2.9. Тест на відмову.

2.2 Нейромережа для генерації воксельної моделі 3D-R2N2

Мережа вивчає зіставлення зображень об'єктів на їх основні тривимірні форми з великої колекції синтетичних даних [17]. Мережа приймає одне або декілька зображень об'єкта з довільних точок зору і виводить реконструкцію об'єкта у вигляді 3D-мережі заповнення. На відміну від більшості попередніх робіт, мережа не вимагає анотацій із зображеннями або міток об'єктів для навчання або тестування.

Швидке та автоматичне прототипування 3D-об'єктів стало нововведенням, що змінюється в грі, у багатьох додатках, пов'язаних із електронною комерцією, візуалізацією та архітектурою. Ця тенденція посилилася зараз, коли 3D-друк – це демократизована технологія, а методи 3D-зйомки є точними та ефективними. Крім того, ця тенденція також поєднується з розповсюдженням масштабних сховищ 3D-об'єктних моделей, таких як ShapeNet.

Однак більшість найсучасніших методів реконструкції 3D-об'єктів підпадають під ряд обмежень. Деякі обмеження полягають у тому, що:

- об'єкти слід спостерігати з великої кількості поглядів; або рівнозначно, перегляди повинні мати порівняно невелику базову лінію. Це питання, коли користувачі бажають реконструювати об'єкт лише з декількох поглядів або в ідеалі лише з одного виду (див. Рис. 2.10 (а));
- очікується, що зовнішність об'єктів (або їх відбиваючих функцій) буде ламбертіанською (тобто невідбиваючою), а альbedo повинен бути неоднорідним (тобто багатим неоднорідними текстурами).

Ці обмеження впливають із ряду ключових технічних припущень. Одним із типових припущень є те, що особливості можуть відповідати поглядам, що є основою більшості методів, заснованих на SFM або SLAM. Було продемонстровано, що якщо точки зору розділені великою базовою лінією, встановлення (традиційних) ознак є вкрай проблематичним через локальні зміни зовнішнього вигляду або самовиключення. Більше того,

відсутність текстури на об'єктах та дзеркальні відображення також ускладнюють проблему узгодження ознак.

Для того, щоб обійти проблеми, пов'язані з великими базовими лініями чи не ламбертійськими поверхнями, стали популярними 3D об'ємні методи реконструкції, такі як різьба по космосу та їх імовірнісні розширення. Однак ці припущення припускають, що об'єкти точно сегментовані від фону або камери відкалібровані, що в багатьох програмах не так.

Інша філософія полягає в тому, щоб припустити, що попередні знання про зовнішній вигляд і форму об'єкта доступні. Перевага використання пріорів полягає в тому, що наступний метод реконструкції менш покладається на пошук точних відповідності характеристик у різних видах. Таким чином, методи, засновані на формуванні, можуть працювати з меншою кількістю зображень і з меншою кількістю припущень щодо функції відбиття об'єкта.

Форми зазвичай кодуються у вигляді простих 3D-примітивів, як це було показано на ранніх піонерських роботах або засвоєних із багатих сховищ 3D-моделей CAD, де концепція пристосування 3D-моделей до зображень обличчя було досліджено значно більшою мірою. Також були введені складні математичні формулювання для адаптації 3D-образних моделей до спостережень з різним ступенем спостереження та різних стратегій регуляризації.

Цей документ у тому ж дусі, що і методи, про які йшлося вище, але з ключовою відмінністю. Замість того, щоб намагатися зіставити відповідну тривимірну форму до спостереження за об'єктом і, можливо, адаптуватися до нього, ми використовуємо глибокі звивисті нейронні мережі, щоб вивчити відображення від спостережень до їхніх тривимірних форм об'єктів із великого набору навчальних даних.

Використовуючи раніше отримані дані, які використовували машинне навчання для вивчення 2D-3D-карти для розуміння сцен, нещодавно запропоновані підходи, спрямовані на обробку даних, щоб вирішити непросту проблему відновлення форми об'єкта лише з одного зображення для задана

кількість категорій об'єктів. Однак у нашому підході ми вперше використовуємо здатність глибоких нейронних мереж автоматично навчатись просто, в кінці кінців, відповідних проміжних уявлень з даних для відновлення наближених реконструкцій 3D-об'єктів з кількох людей, як одного зображення з мінімальним наглядом.

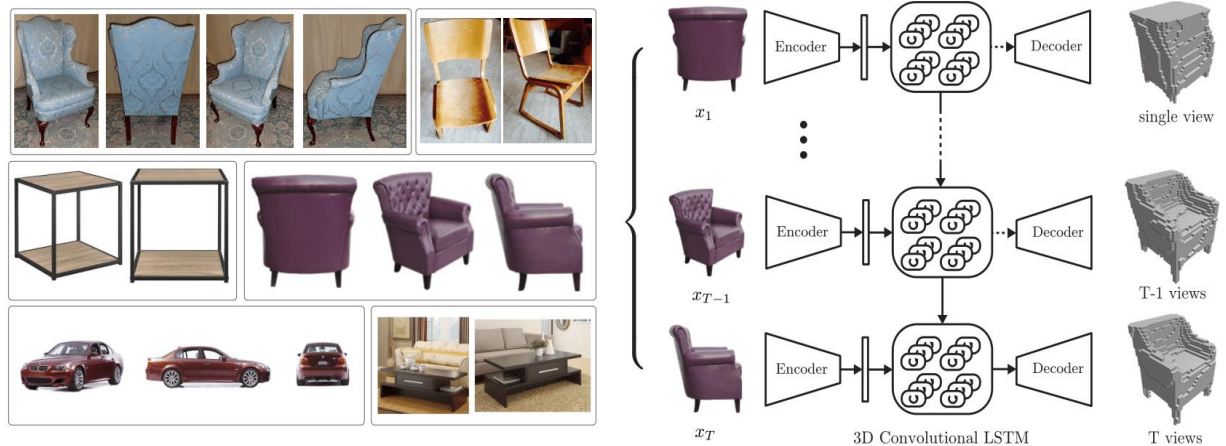


Рисунок 2.10. (а) Деякі зразкові зображення об'єктів, які ми хочемо реконструювати – зауважте, що погляди розділені великою базовою лінією, а зовнішній вигляд об'єктів має невелику текстуру та / або є неамбертійським. (б) Огляд запропонованого нами 3D-R2N2: мережа приймає послідовність зображень (або лише одне зображення) з довільних (некаліброваних) точок зору в якості вхідних даних (у цьому прикладі – 3 виду крісла) та генерує вокселізовану реконструкцію 3D як вихід. Реконструкція поступово відновлюється, оскільки мережа бачить більше переглядів об'єкта.

Натхненний успіхом мереж довготривалої пам'яті (LSTM), а також нещодавнім прогресом у реконструкції 3D-режимів в одному огляді за допомогою згорткових нейронних мереж, ми пропонуємо нову архітектуру, яка називається «нейромережа тривимірної реконструкції» 3D (3D-R2N2). Мережа займає одне ціле або більше зображень екземпляра об'єкта з різних точок зору та виводить реконструкцію об'єкта у вигляді 3D-сітки заповнення, як показано на рис. 2.10 (б). Зауважте, що і в навчанні, і в тестуванні наша

мережа не вимагає ніяких міток об'єктних класів або анотацій зображень (тобто ніякі сегментації, ключові точки, мітки точки огляду або мітки класу не потрібні).

Одним з ключових атрибутів 3D-R2N2 є те, що він може вибірково оновлювати приховані уявлення, контролюючи вхідні ворота та забувати ворота. Навчаючись, цей механізм дозволяє мережі адаптивно та послідовно вивчати відповідне 3D-представлення об'єкта, оскільки (потенційно конфліктна) інформація з різних точок зору стає доступною (див. рис. 2.10). Основні внески цієї статті узагальнені наступним чином:

- Запропоновано розширення стандартної рамки LSTM, яку ми називаємо 3D-періодичною нейронною мережею реконструкції, яка підходить для розміщення каналів зображень з декількома переглядами принципово.
- Об'єднано одно- та багатогранну 3D-реконструкцію в єдину рамку.
- Цей підхід вимагає мінімального нагляду за навчанням та тестуванням (лише обмежувальні коробки, але не потрібна сегментація, ключові точки, мітки точки огляду, калібрування камери або мітки класу).
- Великий експериментальний аналіз показує, що рамка реконструкції перевершує сучасний метод реконструкції одного виду.
- Мережа дозволяє тримати 3D-реконструкцію об'єктів у ситуаціях, коли традиційні методи SFM / SLAM виходять з ладу (через відсутність текстури або широких базових ліній).

Огляд мережі реконструкції показаний на рис. 2.10 (б).

Мережа складається з трьох компонентів: 2D згорткової нейронної мережі (2D-CNN), нової архітектури під назвою 3D Convolutional LSTM (3DLSTM) та 3D «розгорткової» нейронної мережі (3D-DCNN) (див. Рис. 2.11.). З огляду на одне або більше зображень об'єкта з довільних точок зору, 2D-CNN спочатку кодує кожне вхідне зображення x в маломірні характеристики $T(x)$.

Кожен 3D-R2N2 складається з кодера, блоку рецидивування та декодера. Після кожного шару згортання ми розміщуємо нелінійність LeakyReLU. Кодер перетворює зображення 127×127 RGB в низькорозмірну функцію, яка потім подається в 3D-LSTM. Потім декодер приймає приховані стани 3D-LSTM і перетворює їх на остаточну карту заповнення вокселів. Після кожного конвертного шару є Leaky ReLU. Ми використовуємо дві версії 3D-R2N2: (зверху) неглибока мережа та (знизу) глибока залишкова мережа.

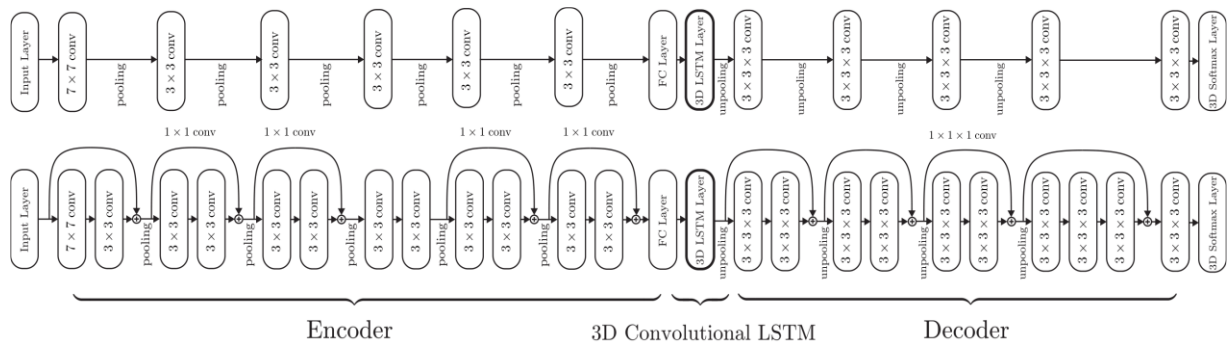


Рисунок 2.10. Архітектура мережі:

Потім, з урахуванням кодованого входу, набір щойно запропонованих тривимірних 3D-згорткових LSTM (3D-LSTM) одиниць (Розділ 3.2) або вибірково оновлює свої клітинні стани, або зберігає стани, закриваючи вхідний затвор. Нарешті, 3D-DCNN декодує приховані стани одиниць LSTM та генерує тривимірну реконструкцію вокселів.

Основна перевага використання мережі на основі LSTM полягає в її здатності ефективно керувати перекриттям об'єктів, коли в мережу подається кілька поглядів. Мережа вибірково оновлює комірки пам'яті, які відповідають видимим частинам об'єкта. Якщо подальший вигляд показує частини, які раніше були самозатіненими, і не відповідали передбаченню, мережа оновила б стани LSTM для раніше прихованих ділянок, але зберегла б стан інших частин.

Ми використовуємо CNN для кодування зображень у функції. Ми розробили два різних кодери 2D-CNN, як показано на рис. 2.11: Стандартний

CNN подачі та глибока залишкова модифікація. Перша мережа складається зі стандартних шарів згортки, об'єднання шарів і непрохідних випрямлених лінійних одиниць з подальшим повністю пов'язаним шаром. Мотивовані нещодавніми дослідженнями, ми також створили глибоку залишкову варіацію першої мережі та повідомили про ефективність цієї зміни в Розділі 5.2. Згідно з дослідженням, додавання залишкових з'єднань між стандартними рівнями згортки ефективно покращує та пришвидшує процес оптимізації для дуже глибоких мереж. Глибока залишкова варіація мережі кодера має з'єднання відображення ідентичності після кожні 2 шари згортання, за винятком 4-ї пари. Для відповідності кількості каналів після згортків ми використовуємо згортку 1×1 для залишкових з'єднань. Потім вихід кодера вирівнюється і передається повністю пов'язаному шару, який стискає висновок у 1024 розмірний особливий вектор.

Основна частина нашого 3D-R2N2 – це модуль запам'ятовування, який дозволяє мережі зберігати побачене та оновлювати пам'ять, коли бачить нове зображення. Наївним підходом було б використання простої мережі LSTM. Однак передбачити такий великий вихідний простір ($32 \times 32 \times 32$) було б дуже важким завданням без будь-якої регуляризації. Ми пропонуємо нову архітектуру, яку ми називаємо 3D-Convolutional LSTM (3D-LSTM). Мережа складається з набору структурованих блоків LSTM з обмеженими з'єднаннями. Блоки 3D-LSTM просторово розподіляються у структурі 3D-сітки, кожен блок відповідає за реконструкцію певної частини кінцевого виводу. Всередині 3D-сітки знаходяться $N \times N \times N$ 3D-LSTM-одиниць, де N – просторове дозвіл сітки 3D-LSTM.

Позначимо операцію згортання як $*$. У нашій реалізації ми використовуємо $N = 4$. На відміну від стандартного LSTM, у нас немає вихідних воріт, оскільки ми отримуємо лише вихід у кінці. Видаляючи надлишкові вихідні ворота, ми можемо зменшити кількість параметрів.

Інтуїтивно ця конфігурація змушує блок 3D-LSTM вирішувати невідповідність між певним регіоном передбачуваної реконструкції та

базовою моделлю істини таким чином, що кожен блок вчить реконструювати одну частину воксельного простору, а не сприяти реконструкції всього простору. Ця конфігурація також наділяє мережу почуттям локальності, щоб вона могла вибірково оновлювати своє передбачення щодо раніше закритої частини об'єкта. Ми візуалізуємо таку поведінку в додатку.

Більше того, 3D-згортковий LSTM-блок обмежує з'єднання прихованого стану до просторових сусідів. Для ванільних LSTM всі елементи в прихованому шарі $h_t - 1$ впливають на поточний прихований стан h_t , тоді як просторово структурований 3D згортковий LSTM дозволяє лише на його приховані стани $h_t((i, j, k))$ впливати на сусідній 3D-LSTM одиниці для всіх i, j та k . Більш конкретно, сусідні з'єднання визначаються розміром ядра згортки. Наприклад, якщо ми використовуємо ядро $3 \times 3 \times 3$, на блок LSTM впливають лише його безпосередні сусіди. Таким чином, блоки можуть розподіляти ваги, а мережу можна додатково регулювати. У розділі 2 ми також описали рецидивіруему одиницю (GRU) як варіацію одиниці LSTM. Ми створили варіацію 3D-згорткового LSTM за допомогою рецидивованого блоку (GUT).

Після отримання послідовностей вхідних зображень x_1, x_2, \dots, x_T , 3D-LSTM передає h_T стану прихованого стану декодеру, що збільшує роздільну здатність прихованого стану, застосовуючи 3D згортки, нелінійності та 3D розкручування поки не досягне цільової вихідної роздільної здатності.

Як і у випадку з кодерами, ми пропонуємо просту декодерну мережу з 5 згортками та глибоку залишкову версію з 4 залишковими з'єднаннями з подальшим остаточним згортанням.

Розширення даних: Навчаючи, ми використовували 3D CAD-моделі для генерування вхідних зображень та наземних карт зайнятості вокселів. Ми спочатку надали моделі CAD з прозорим фоном, а потім доповнили вхідні зображення випадковими культурами з набору даних PASCAL VOC 2012. Також ми підфарбовували колір моделей і випадковим чином перекладали зображення. Зауважте, що всі точки огляду були відібрані вибірково.

Навчання: Під час тренінгу в мережі ми використовували входи змінної довжини від одного зображення до довільної кількості зображень. Більш конкретно, довжина введення (кількість переглядів) для кожного прикладу тренінгу в межах однієї міні-партії зберігалася постійною, але тривалість введення прикладів тренувань у різних міні-партіях змінювалася випадковим чином. Це дало змогу мережі здійснити як одно-, так і багато-оглядну реконструкцію. Під час тренінгу ми обчислювали втрати лише в кінці послідовності введення, щоб заощадити як обчислювальну потужність, так і пам'ять. З іншого боку, під час тестування ми могли отримати доступ до проміжних реконструкцій на кожному етапі часу, витягуючи приховані стани одиниць LSTM.

Мережа: розмір вхідного зображення встановлено на 127×127 . Реконструкція вихідного вокселізованого розміру була розміром $32 \times 32 \times 32$. Мережі, які використовувались в експериментах, пройшли навчання 60 000 ітерацій з розміром партії 36, за винятком [Res3D-GRU3], для розміщення в GPU NVIDIA Titan X потрібен розмір партії 24. Для шарів LeakyReLU нахил витоку встановлювався на всій мережі 0,1. Для розгортки ми дотримувались схеми відмовлення, представленої в інших роботах. Ми використовували Theano для реалізації нашої мережі та використовували оптимізатор Адама для правила оновлення SGD.

ShapeNet: Набір даних ShapeNet – це набір 3D-моделей CAD, які організовані відповідно до ієрархії WordNet. Ми використовували підмножину набору даних ShapeNet, яка складається з 50 000 моделей та 13 основних категорій. Ми розділили набір даних на набори для навчання та тестування, 4/5 – для тренінгу, а решту – 1/5 для тестування. Ці два набори даних ми називаємо навчальним набором ShapeNet та тестовим набором у розділі експериментів.

PASCAL 3D: Набір даних PASCAL 3D складається з зображень виявлення PASCAL 2012, доповнених 3D-CAD-моделлю вирівнювання.

Інтернет-продукти: Набір даних містить зображення 23 000 предметів, що продаються в Інтернеті. Методи MVS та SFM виходять з ладу на цих зображеннях через надзвичайно широкі базові лінії. Оскільки в наборі даних немає наземних 3D-моделей CAD, ми використовували набір даних лише для якісної оцінки.

Моделі CAD MVS: Для порівняння методу з методом Multi View Stereo ми зібрали 4 різні категорії високоякісних моделей САПР. Усі CAD-моделі мають багато текстурних поверхонь і розміщуються поверх паперу, насиченого текстурою, щоб сприяти локалізації камери методу MVS.

Показники. Використовували два показники для оцінки якості реконструкції. Основним показником було перетину вокселів з вихідними даними (IoU) між реконструкцією вокселів 3D та його основної моделлю дійсної вокселізації.

2.3 Алгоритм знаходження параметрів об'єктів

Сітки переважно використовуються для представлення 3D-моделей, оскільки вони добре поєднуються з обладнанням для графічного відтворення. Такі функції, як об'єм, моменти та коефіцієнти перетворення Фур'є, повинні ефективно обчислюватися із представлення сітки. Щоб обчислити особливість для сітки, ми покажемо, що спочатку ми можемо обчислити її для кожної елементарної форми, наприклад трикутника або тетраедра, а потім скласти всі значення для сітки. Алгоритм простий і ефективний, з багатьма потенційними можливостями для розширення[19].

Обчислення обсягу 3D-моделі не є тривіальною роботою. Можна перетворити модель у дискретний 3D-двійкове зображення. Точки сітки в дискретному просторі називаються вокселями. Кожен воксель позначений позначкою «1» або «0», щоб вказати, чи знаходиться ця точка всередині або поза об'єктом. Кількість вокселів всередині об'єкта, або еквівалентно підсумовування всіх значень вокселів у дискретному просторі, може бути

приблизною до обсягу моделі. Однак перетворення з 3D-сітчастої моделі у бінарне зображення дуже трудомістке. Крім того, для підвищення точності дозволу 3D-бінарного зображення потрібно бути дуже високим, що може додатково збільшити навантаження на обчислення.

Пояснюємо наш підхід, починаючи з обчислення площ для 2D сіток. 2D сітка – це просто 2D форма з полігональними контурами. Як показано на малюнку 1, припустимо, у нас є 2D сітка із жирними лініями, що представляють її краї. Хоча ми можемо розрізнити 2D простір на двійкове зображення та обчислити площу сітки, підрахувавши пікселі всередині багатокутника, але це дуже просто обчислювати.

Для початку з нашого алгоритму, зробимо припущення, що багатокутник близький. Якщо його немає, спочатку можна виконати процес контурного пошуку. Оскільки знаємо всі вершини та ребра многокутника, ми можемо легко обчислити норму для кожного ребра.

Тут ми визначаємо нормальне як нормалізований вектор, який перпендикулярний відповідному краю і спрямований назовні від сітки. У літературі з комп'ютерної графіки існують різні способи перевірити, чи знаходиться точка всередині або поза полігоном, таким чином, легко знайти правильний напрямок нормалей. Пізніше ми покажемо, що навіть якщо ми лише знаємо, що всі норми вказують на одну і ту ж сторону сітки (всередині або зовні, до тих пір, поки вони відповідають), ми все одно зможемо знайти правильну площу сітки.

Отримавши нормалі, побудуємо набір трикутників, з'єднавши всі вершини багатокутника з початком.

Кожне ребро і початок утворюють елементарний трикутник, який є найменшою одиницею для обчислення. Ми визначаємо підписану область для кожного елементарного трикутника, як показано нижче: Величина цього значення – це площа трикутника, тоді як знак значення визначається шляхом перевірки положення початку по відношенню до краю та напрямку нормалі.

Ми можемо розширити вищевказаний алгоритм на тривимірний випадок. У файлі VRML сітка представлена набором вершин і полігонів. Перш ніж обчислити обсяг, робимо деяку попередню обробку на моделі та переконуємось, що всі багатокутники є трикутниками. Така попередня обробка, яка називається тріангуляцією, зазвичай використовується в кодуванні сітки, обробці сітчастого сигналу та редагуванні сітки. Напрямок нормалі для трикутника можна визначити за порядком вершин і правим правилом, як показано на малюнку 2. Послідовну умову дуже легко виконати. Для двох сусідніх трикутників, якщо спільний край має різні напрямки, то норми двох трикутників узгоджуються. Наприклад, АВ – спільний край трикутника АСВ і АВD. У трикутнику АСВ напрям – від В до А, а в трикутнику АВD – напрямок від А до В, таким чином N_{ACB} і N_{ABD} є послідовними.

У тривимірному випадку елементарною одиницею обчислення є тетраедр. Для кожного трикутника ми з'єднуємо кожен його вершину з початком і утворюємо тетраедр, як показано на малюнку 3. Як і у випадку 2D, ми визначаємо підписаний об'єм для кожного елементарного тетраедра як: Величина його значення – це об'єм тетраедра, і знак значення визначається, перевіряючи, чи походження стоїть на тій же стороні, що і нормальна щодо трикутника. Обсяг 3D-сітчастої моделі завжди позитивний. Кінцевого результату можна досягти, взявши абсолютне значення V' для усього. Для обчислення інших функцій 3D-моделі, таких як моменти або коефіцієнти перетворення Фур'є, ми повертаємо послідовність вершин для кожного трикутника, якщо сумарна V' виявляється негативною.

Висновки по розділу

У даному розділі детально розглянуто архітектури нейромереж для створення тривимірних моделей з зображень, описаний процес взаємодії роботизованого пристрою, з наведенням конкретних прикладів. Для розробки інтерфейсу користувача обрано застосовувати Visual Studio Code. Описаний набір даних для навчання, детальний процес його формування, обґрунтування вибору, описано перелік датасетів, які можна використовувати. Проаналізовані нейронні мережі, які генерують каркасну сітку та вокселі, та обрано оптимальну з існуючих. Описаний процес навчання.

РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Архітектура програмного забезпечення

Основними завданнями для експериментальної частини буде реалізація системи розпізнавання характеристик, щоб перевірити, наскільки добре працює детектор, навчений загальними даними у конкретній задачі, а також потенційні методи для поліпшення.

На додаток до обчислювальної потужності, глибокі згорткові мережі вимагають великої кількості навчальних даних. Оскільки збирання та анотування набору даних є витратним, більшість досліджень виконуються на декількох доступних наборах даних.

Загальний детектор об'єктів повинен бути навчений різним даним і підтримувати кілька категорій об'єктів. Стандартні набори тестів (PASCAL VOC та ImageNet) забезпечили відповідну відправну точку. Об'єктивний детектор об'єктів також повинен бути легко доступним і простим у застосуванні. Існує безліч доступних мереж, які були попередньо підготовлені в стандартних наборах даних.

Для оцінки ефективності пошуку об'єктів за конкретною задачею ми зібрали дані тесту з різних джерел. Основний критерій полягав у тому, що об'єктові анотації даних тесту повинні бути приблизно сумісними з об'єктом анотацій стандартних тестів. Це означає, що тестові набори даних повинні мати спільні класи з стандартними орієнтирами та включати в себе анотації об'єктів як обмежувальні поля.

Навіть незважаючи на те, що передові методи, такі як Hourglass, забезпечують незначне підвищення точності, основним їх внеском є покращення швидкості. Оскільки оцінка часу виконання є важливою характеристикою для робототехнічних систем, ці методи мають додаткову цінність для експериментів.

Було використано попередньо навчену мережу, структура якого показана на демонстраційному плакаті №4. Більшість операцій мереж показані в вигляді окремих шарів. повністю згорткова підмережа, як показано в лівому верхньому кутку.

Стандартні орієнтири, що використовуються для виявлення об'єктів, є набір даних для завдання PASCAL 3D+ і набір даних для розпізнавання ImageNet у великих масштабах.

Основним набором для навчання був пакет даних PASCAL 3D+. У 2007 році набір даних містив 9,963 анотованих зображень з 20 класами об'єктів (людина, 6 різних тварин, у тому числі автомобілів). На наступний рік набір даних збільшився до 54,9 тис. Зображень, близько половини з яких склали тестові та перевірочні набори. Кількість і тип різних завдань також різняться щороку.

Передчасна мережа PASCAL VOC, яку була використана, сама по собі була навченої на наборі даних ImageNet. Починаючи з 2013 року, проблема включала завдання виявлення об'єктів, в якій в даний час містяться 200 повністю помічених категорій (включаючи ті ж самі категорії, що і PASCAL VOC). Навчальний комплект складається з 456 567 зображень, а набір для валідації складається з 21121 зображень.

Зараз же розберемо архітектуру програмного забезпечення на якому вона побудована. Для кращого розуміння візьмемо конкретний випадок:

1. Надання інформації. Клієнт отримує зображення об'єкта використовуючи готове зображення або робить новий знімок камерою.
2. Конвертація даних. У залежності від того через від способу введення необхідними є відповідні апаратні та програмні засоби.
3. Після початкового апаратного рівня інформація передаються до програми, за допомогою якої здійснюється розпізнавання. Це можуть бути додатки типу WEB, мобільних, або на будь-якій платформі ОС. У нашому випадку, набагато вигіднішим є інтеграція в вже існуючі додатки для роботи з зображеннями, так як є це гарантує прийняття нововведень користувачами,

позбавляє необхідності встановлення, або налаштування та є дуже економічно вихідним.

4. Далі йде передача даних на фізичний сервер, на якому і знаходиться головна API, або який є зв'язковою ланкою для обміну даними. Перший варіант є пріоритетним, але складним в реалізації, так як у цьому випадку прискорюється швидкість обробки інформації, але значно зростають вимоги до апаратного забезпечення клієнта, що є одним з головних параметрів при розробці подібних систем.

5. У самій API власне й знаходиться майже вся логіка програми. З точки зору апаратної реалізації нас цікавить взаємодія з іншими пристроями, і вона визначається саме тут, при обробці зображення.

6. Ці модулі можуть бути найрізноманітнішими і залежать часто від конкретного спектру задач, які потрібно виконати. Це може бути й виконання якихось механічних дій, наприклад, у випадку з роботом, що буде переміщуватися повністю, або рухати якісь частини.

Процес користування є простим та інтуїтивно зрозумілим для кожного. Він може трішки відрізнятись у залежності від платформи в яку він інтегрований, але ці зміни є зовсім не значними.

Моделлю на, якій функціонує нейронна мережа є модель згорткової нейромережі. Вона є основою для функціонування системи. Задачею є визначення параметрів, при яких функціонування мережі буде найоптимальнішим.

Першим досліджуваним параметром був — кількість зображень, які входять у навчальний набір для кожного класу об'єктів. Дослідження проводиться для моделі яка навчена на 5000 зображеннях.

Дослідження різних параметрів показало, що при малому наборі даних для навчання точність погіршується через недостатнє розуміння класів та варіантів ключових точок в них, а при надмірній кількості зображень відбувається перенавчання. Було визначено, що для моделі Hourglass оптимальними параметрами є: ефект складання модулів пісочного годинника

разом та вплив проміжного нагляду. Вони взаємно не залежать, оскільки ми обмежені в тому, як ми можемо застосовувати проміжний нагляд залежно від загального архітектурного дизайну. Застосовуючись окремо, кожен позитивно впливає на продуктивність, і ми разом бачимо подальше вдосконалення швидкості тренувань і, врешті-решт, остаточну оцінку ефективності пози.

3.2 Опис структури тестової моделі

Розпізнавання об'єктів, таких як пішохід та виявлення транспортних засобів, є популярними темами дослідження у комп'ютерному зору. Такі об'єкти анотовані в багатьох публічно доступних збірках даних. Це дало джерело даних для тестування загального детектора об'єктів. Автомобілі та люди також анотовані в тестових наборах даних, забезпечуючи сумісність[20].

Перший набір тестів, який ми використовували, є тим самим набором даних, який використовувався в публікації Putting Objects in Perspective (OiP). Набір містить 600 тестових зображень (з яких 422 включають допустимі об'єкти) та 60 вибраних зображень встановлених валідацій з бази даних LabelMe. Об'єкти, що входять до зображень, були позначені обмежувальними полями. Ми використали 60 зображень валідації як нашого першого тестового набору, до складу якого входять 126 легкових автомобілів та 84 пішоходів.

Є два класи об'єктів: автомобілі (у тому числі регулярні легкові автомобілі, а також фургони та вантажні автомобілі) та пішоходів (вертикальні особи). Набір даних «PASCAL VOC» містить однакові категорії. Транспортні засоби набору даних PASCAL включають (пасажирські) автомобілі та автобус і як окремі класи, але ніяких інших чотириколісних видів транспорту. Проте, це розумна гіпотеза про те, що детектор, навчений цим двом класам, також навчиться розпізнавати більш загальний автомобільний клас набору даних OiP. Інший клас OiP, пішохід, суворо кажучи, є підкласом категорії людей PASCAL. Однак, розумно припустити, що більшість людей на зображенні вулиці є пішоходами.

Набір даних Street Scenes включає в себе 3547 зображень, які були марковані для 9 різних об'єктів (автомобіль, пішохід, велосипед, будівля, дерево, дорога, небо, тротуар та магазин). Було використали 60 зображень, обраних з цього набору, як наш другий набір тестів. Ця частина набору включає 131 машину та 51 пішохід.

Розташування об'єктів анотовано за допомогою багатокутників, які тісно оточують об'єкт. Полігони можуть бути перетворені в обмежувальну область з урахуванням мінімальних та максимальних значень для координат x і y . Клас автомобіля включає в себе всі моторизовані транспортні засоби (з більш ніж двома колесами) шириною більше 64 пікселів. Пішоходи включають осіб, які ходять або стоять і мають висоту більше 32 пікселів. Обидва об'єкти позначаються лише якщо їх видима частина складає 75%.

Був створений скрипт Python для виклику функцій CNN та для виконання попередньої обробки, яка надається TensorFlow. Сценарій починається, завантажуючи мережу. Мережа вводиться в тестовий режим, що вимикає навчання. Граничні поля потім перетворюються в формат координат, який використовується мережею. Далі зображення та коробки масштабуються відповідно до розміру входу в мережі (600x800 пікселів з 3 кольоровими каналами). Зображення також попередньо обробляються шляхом віднімання середнього значення кольору.

Після підготовки даних функція оцінки мережі викликається з використанням зображення та області як вхідних аргументів. Після оцінки, класові ймовірності та дельта-значення для кожної області знімаються "поза мережею". Далі рецензії обмежувальної копії обчислюються шляхом об'єднання вихідних значень та значень дельти. а також ймовірності (автомобілі, автобуси та особи) зберігаються в структурі. Замовлення автомобілів і автобусів поєднуються в оцінці для створення наближення більш загальноприйнятого класу автомобілів, який включає в себе великі автомобілі, а також легкові автомобілі. Виявлення зберігаються базі даних.

3.3 Вимоги до технічного забезпечення

Основними проблемами, що обмежують використання глибоких методів навчання, є нестача продуктивності комп'ютера для навчання. Для цієї дисертації ми не мали доступу до серверної або високопродуктивного графічного процесора. Швидше, нам довелося реалізовувати методи на серійному ноутбуці. Навчання згорткової мережі від початку до кінця на такому обладнанні буде надзвичайно довгим. Таким чином, ми віддали перевагу таким методам, які були встановлені достатньо для того, щоб мати доступ до попередньо підготовлених мереж.

Основним обладнанням для реалізації нейронної мережі був використаний комп'ютер з такими характеристиками:

- Процесор: Intel Core i5-2500k
- Кількість ядер: 4
- Базова частота: 3,30 GHz
- Відеокарта: NVIDIA GeForce GTX 560
- Оперативна пам'ять: HyperX DDR4-2133 8192MB
- Жорсткий диск: Western Digital Blue 1TB 7200rpm
- Операційна система: Windows 10.
- Основне програмне забезпечення Tensorflow 1.7.0.

Для простого клієнта який використовує API достатньо:

- процесор – 1.6 ГГц, 1 ядро ЦП або краще;
- оперативна пам'ять не менш ніж 1024 Мб;
- не менше ніж 35 ГБ ПЗУ;
- доступ до мережі Інтернет;
- Операційна система: Linux, Mac OS, Windows, and others;

Для коректної роботи серверної частини додатку має використовуватися машина з такими мінімальними характеристиками:

- Процесор з частотою 1.8 ГГц;
- Об'єм оперативної пам'яті 4 ГБ;

- HDD або SSD накопичувач з об'ємом 50 Гб;
- Доступ до мережі Інтернет.

Виходячи із специфіки роботи додатку для коректної роботи клієнтської частини необхідні такі мінімальні характеристики:

- Встановлений браузер Internet Explorer версії 10 або вище;
- Встановлений браузер Google Chrome, Opera, Mozilla Firefox, або інші браузери, що використовують той самий движок;
- Наявність підключення до мережі Інтернет;
- Наявність камери.

3.4 Вимоги до програмного забезпечення

Було реалізовано згорткову мережу, використовуючи інструментарій Python та TensorFlow.

TensorFlow – це бібліотека з відкритим кодом для машинного навчання та високоефективних чисельних обчислень. Її обмежена архітектура дозволяє без проблем розгортати обчислення через різні платформи (процесори, графічні процесори, TPU), а також від настільних комп'ютерів до кластерів серверів до мобільних та крайових пристроїв. TensorFlow спочатку був розроблений дослідниками та інженерами з команди Brain Google в рамках П-організації Google, він надає потужну підтримку машинного навчання та глибокого навчання.

Остання версія TensorFlow становить 1.7.0 та доступна на офіційному веб-сайті TensorFlow www.tensorflow.org. Він був розроблений з урахуванням глибокого навчання, але застосовний до набагато ширшого кола проблем.

Він може бути розгорнутий у широкому діапазоні продуктів Google, таких як розпізнавання мови, Google Photos, Gmail і навіть Пошук за допомогою процесорів та графічних процесорів, а також можна обіграти ідею на своєму ноутбукі, не маючи іншого обладнання.

Ви повинні виразити свій розрахунок у вигляді діаграми потоку інформації, щоб використовувати TensorFlow. Це глибоко пристосований каркас, який дає можливість одночасно подавати різні моделі або різні передачі подібної моделі. Техніка TensorFlow дуже зважена, що означає, що ви можете використовувати кілька секцій самостійно або використовувати кожну з частин разом.

TensorFlow дає максимальну користь від вашого доступного обладнання за допомогою просунутої допомоги, неперевершеного розрахунку та ліній. Просто призначте фігурним компонентам вашої діаграми TensorFlow різним пристроям і дозвольте їй вирішувати самі дублікати. Це також заохочує вас з вибором діалекту для виконання вашої обчислювальної діаграми. Блокнот іPerson TensorFlow допомагає зберігати коди, нотатки та представлення у послідовно зібраному та інтуїтивно зрозумілому стилі.

Як основний програмний засіб для розробки був обраний TensorFlow. Є ряд причин чим це зумовлено:

1. Простота побудови моделі. TensorFlow пропонує кілька рівнів абстракції, щоб ми могли вибрати правильний для наших потреб. Дає можливість створювати та навчати моделі за допомогою API Keras високого рівня, що робить роботу з TensorFlow та машинним навчанням досить простою.

2. Надійна продукція ML в будь-якому місці. TensorFlow завжди забезпечував прямий шлях до виробництва. Будь то на серверах, периферійних пристроях або в Інтернеті, TensorFlow дозволяє легко навчати та розгортати модель, незалежно від мови та платформи, якими ви користуєтесь. Для отримання повного конвеєру ML існує TensorFlow Extended (TFX). Для виконання на мобільних та периферійних пристроях існує TensorFlow Lite. Для розгортання моделей в середовищах JavaScript, використовують TensorFlow.js.

3. Потужний засіб для досліджень. Створення та тренування найсучасніших моделі, не жертвуючи швидкістю та продуктивністю.

TensorFlow надає гнучкість та контроль за допомогою таких функцій, як Keras Functional API та API субкласифікації моделей для створення складних топологій. TensorFlow також підтримує екосистему потужних додаткових бібліотек та моделей, з якими можна експериментувати, включаючи Ragged Tensors, TensorFlow Probability, Tensor2Tensor та BERT.

Іншим програмним засобом є Keras. API нейронних мереж високого рівня, написаний на Python і здатний працювати над TensorFlow, CNTK або Theano. Він був розроблений з акцентом на швидке експериментування. Вміння переходити від ідеї до результату з найменшою можливою затримкою є ключовим для хорошого дослідження.

Це бібліотека глибокого навчання, яка:

- Дозволяє легко та швидко прототипувати (завдяки зручності користування, модульності та розширюваності).
- Підтримує як зготкові мережі, так і періодичні мережі, а також їх комбінації.
- Працює безперебійно на процесорі та GPU.

TensorFlow Lite – це набір інструментів, які допомагають розробникам запускати моделі TensorFlow на мобільних, вбудованих та IoT-пристроях. Це дає змогу машинному навчанню робити висновок із низькою затримкою та невеликим бінарним розміром.

TensorFlow Lite складається з двох основних компонентів:

- Перекладач TensorFlow Lite, який працює спеціально оптимізовані моделі на багатьох різних типах обладнання, в тому числі мобільних телефонів, вбудованих пристроїв Linux і мікроконтролерах.
- Конвертер TensorFlow Lite, який перетворює модель TensorFlow в ефективну форму для використання інтерпретатора, і може ввести оптимізацію, щоб поліпшити бінарний розмір і продуктивність.

TensorFlow Lite розроблений так, щоб полегшити машинне навчання на пристроях "на краю" мережі, а не надсилати дані туди-сюди з сервера. Для

розробників виконання машинного навчання на пристрої може допомогти покращити:

- Затримка: немає зворотного шляху до сервера.
- Конфіденційність: жодних даних не потрібно залишати з пристроєм.
- Підключення: підключення до Інтернету не потрібно.
- Споживання електроенергії: мережеві з'єднання не потребують живлення.

TensorFlow Lite працює з величезним набором пристроїв, від крихітних мікроконтролерів до потужних мобільних телефонів.

Ключові риси:

- Інтерпретатор налаштований на ML на пристрої, підтримуючи набір основних операторів, оптимізованих для додатків на пристрої, і з невеликим бінарним розміром.
- Підтримка різноманітних платформ, що охоплює пристрої Android та iOS, вбудований Linux та мікроконтролери, використовуючи API платформи для прискореного виводу.
- API для декількох мов, включаючи Java, Swift, Objective-C, C++ та Python.
- Висока продуктивність, з апаратним прискоренням на підтримуваних пристроях, оптимізованими пристроями ядрами та попередньо сплавленими активаціями та упередженнями.
- Інструменти оптимізації моделей, включаючи квантування, дозволяють зменшити розмір та збільшити продуктивність моделей, не втрачаючи при цьому точності.
- Ефективний формат моделі, використовуючи FlatBuffer, оптимізований для невеликих розмірів і портативності.
- Попередньо підготовлені моделі для поширених завдань машинного навчання, які можна налаштувати під вашу програму.
- Зразки та навчальні посібники, які показують, як розгорнути

моделі машинного навчання на підтримуваних платформах.

Робочий процес використання TensorFlow Lite включає наступні кроки:

- Виберіть модель. Візьміть власну модель TensorFlow, знайдіть модель в Інтернеті або виберіть модель з наших попередньо підготовлених моделей, щоб відмовитися або перевчитись.
- Перетворимо модель. Якщо ви використовуєте власну модель, використовуйте конвертер TensorFlow Lite та кілька ліній Python для перетворення його у формат TensorFlow Lite.
- Розгорніть на своєму пристрої. Запустіть свою модель на пристрої за допомогою інтерпретатора TensorFlow Lite з API на багатьох мовах.
- Оптимізуйте свою модель. Використовуйте наш Інструментарій оптимізації моделей, щоб зменшити розмір вашої моделі та підвищити її ефективність при мінімальному впливі на точність.

TensorFlow Lite планує забезпечити високу продуктивність на пристрої для будь-якої моделі TensorFlow. Однак інтерпретатор TensorFlow Lite в даний час підтримує обмежений підмножина операторів TensorFlow, оптимізованих для використання на пристрої. Це означає, що для роботи з TensorFlow Lite для деяких моделей потрібні додаткові кроки.

Якщо ваша модель використовує операторів, які ще не підтримуються інтерпретатором TensorFlow Lite, ви можете використовувати TensorFlow Select для включення операцій TensorFlow у збірку TensorFlow Lite. Однак це призведе до збільшення бінарного розміру.

TensorFlow Lite наразі не підтримує навчання на пристроях, але воно є в нашій дорожній карті разом з іншими запланованими удосконаленнями.

Висновки по розділу

У даному розділі детально описано архітектуру роботи реалізованої системи отримання характеристик об'єктів по їх зображеннях, реалізовано інтерфейс користувача та описано базу даних, описані етапи роботи алгоритмічного забезпечення, визначені вимоги до технічного забезпечення, описані програмні засоби, що використовувалися. Зображено процес взаємодії з користувачем та інтерфейс. Підведені підсумки по результатам проведеної роботи.

Додатково проведено тест з використанням воксельної та сіткової нейромережі, за результатами експерименту доведено доцільність використання нейронних мереж для задачі перетворення зображення об'єкта в тривимірну модель.

РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту

Ідея проекту полягає у створенні інтелектуальної системи обробки зображень. Призначення системи – обробка створених знімків в додатку та отримання характеристик об'єктів на зображенні. А саме:

- виділення об'єктів створення його 2D та 3D обмежуючої рамки;
- калібрування додатку для отримання реальних лінійних розмірів об'єктів на зображенні;
- створення 3D-моделі об'єктів на зображенні та отримання їх реальних лінійних розмірів;
- відображення статусу доступності хмарного обчислення.

Таблиця 4.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Робочий додаток на мобільний пристрій завдяки якому є можливість фотографувати об'єкти і отримувати їх тривимірні моделі. Додаток підтримує технологію доповненої реальності.	Генерація тривимірних моделей	Не потрібно довге та дороге 3D-сканування. Швидке прототипування.
	Оцінка параметрів об'єктів	Аналіз отриманої 3D-моделі та її лінійні характеристики. Можливість отримати справжні 3D координати.
	Доповнена реальність	Отримані об'єкти можна переміщати та змінювати їх в доповненій реальності.

На ринку існують аналоги подібних систем, але більшість з них налаються з перенасиченим функціоналом, направлені на великі підприємства. Ці аналоги в основному англомовні, дорогі, вимагають

довгостроково навчання персоналу або платне обслуговування. До того ж розроблена система універсальна та може бути пристосована до різних підприємств. Тому доцільно проводити аналіз потенційних техніко-економічних переваг ідеї порівняно з пропозиціями конкурентів.

Таблиця 4.2. Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї
1.	Швидкість роботи
2.	Зручність використання
3.	Вимоги до системи
4.	Кросплатформність

Таблиця 4.3. Головні конкуренти

№	Продукція конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
	Мій проект	Film3D	AutoCAD mobile app			
1	Швидка	Середня	Повільна	Зв'язок з мережею	Простота архітектури	Потрібно менше знань для виконання заданої задачі
2	Зручно	Зручно	Зручно	Для здійснення серії знімків потрібно використовувати штатив	Розвиток UI має перспектив у наявність API	Автоматизована робота

Закінчення таблиці 4.3.

3	Мінімал ьні	Мінімаль ні	Високі	Відсутня оптимізація зі старими системами	Простота системи	Актуальність програми для нових систем через використання новітніх бібліотек
4	Наявна	Відсутня	Відсутня	Відсутня оптимізація для роботи через мобільні пристрої	Налаштована система для роботи через головні інструменти використання користувачам и	—

Ідея проекту є актуальною, можна виділити вагомі переваги для споживачів системи. Перелічені техніко-економічні характеристики, слабкі, нейтральні та сильні сторони дають підставу вважати, що проект може мати успіх.

4.2 Технологічний аудит ідеї проекту

Для проведення технічного аудиту ідеї проекту, потрібно провести аудит технологій, за допомогою яких можна реалізувати ідею проекту. І для початку потрібно визначити можливість технологічної здійсненності проекту.

Обрана технологія доступна, не потребує доробки, а також безкоштовна та надає усі необхідні можливості для реалізації поставленої задачі. Для розробки з використанням даної технології необхідно мати персональний комп'ютер для можливості встановлення робочого середовища.

Таблиця 4.4. Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
1	Видалення рухомих об'єктів з серії знімків	Використання згорткової нейронної мережі	Hourglass	Повністю відкрита для роботи з інформацією
2	Робота безпосередньо з інформацією, яку надає користувач	API для роботи з обробки зображень	Розроблені бібліотеки для роботи нейронної мережі	Повністю відкритий програмний код та використання готових бібліотек
Обрана технологія реалізації ідеї проекту: 1				

Висновок: технологічна реалізація продукту – можлива, вибрана технологія №1 яка може нам допомогти розробити якісний продукт з використанням комбінації технологій, та перспективи у майбутніх ідей.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів. Для цього спочатку проводиться аналіз попиту.

Висновок: враховуючи кількість головних гравців по ринку, зростаючу динаміку ринку, невелику кількість конкурентів та середню норму рентабельності можна зробити висновок, що на даний момент, ринок для входження стартап-продукту є привабливим.

Таблиця 4.5. Попередня характеристика потенційного ринку

№	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	4
2	Загальний обсяг продаж, грн./ум. од	200
3	Динаміка ринку	Темпи розвитку світової економіки позитивні, але з ознаками зменшення росту.
4	Наявність обмежень для входу	Відсутні. Так як аналоги використовують не автоматизовані алгоритми обробки фотографій
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні.
6	Середня норма рентабельності в галузі або по ринку, %	70

Надалі визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи (табл. 4.6). Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають.

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (табл. 4.7) за моделлю п'яти сил М. Портера, яка вирізняє п'ять основних факторів, що впливають на привабливість вибору ринку з огляду на характер конкуренції:

- конкурент, що вже є у галузі;
- потенційні конкуренти;
- наявність товарів-замінників;
- постачальники, що конкурують за ринкову владу;
- споживачі, які конкурують за ринкову владу.

Таблиця 4.6. Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці цільових груп клієнтів	Вимоги споживачів до товару
1	Потреба в зменшенні витрат в часі та батареї телефону	Туристи	Стартап буде переважно буде економити велику кількість часу та витрат заряду батареї	Зручність у використанні. Швидка робота системи. Спроможність швидко освоїти як користуватися системою.
2	Потреба в зменшенні витрат в часі та швидкої публікації фотографій	Звичайні користувачі	Стартап буде переважно буде економити кількість часу та швидкої публікації фотографій в соціальні мережі	Можливість редагувати не вірні результати роботи нейронної мережі.

Визначена характеристика дозволяє зробити висновок, що проект знайде свого покупця, а за рахунок подальшого масштабування та адаптації проект може стати універсальним рішенням.

Таблиця 4.7. Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренти	Наявність конкурентів, котрі надають схожі рішення	Зменшення ціни на поставлену послугу; Розробка унікальних характеристик товару; Надання ліцензій на обслуговування
2	Кошти на розробку та підтримку продукту	Закінчення грошей та недостатнє фінансування	Залучення додаткових інвесторів, мотивація роботи на перспективу; Ітеративна розробка продукту задля покрокового виведення продукту на ринок та отримання відповіді користувачів

Закінчення таблиці 4.7.

3	Вихід аналогу	Вихід аналогу даного товару може призвести до знецінення та безідейності даного товару	Вихід товару на ринок в коротші строки з не повною, але достатньою, функціональністю для зацікавлення усіх цільових аудиторій; Проведення рекламної компанії
---	---------------	--	---

Таблиця 4.8. Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Новий продукт	Вихід на ринок, Зменшення монополії, Надання нових рішень у сфері	Розробка нової функціональності; Вихід нової продукції на ринок; Надання різноманітних типів ліцензій в залежності від потреб користувача \ замовника.
2	Вихід аналогу	Надати продукт з певними характеристиками та можливостями що відсутні у компаній конкурентів	Аналіз ринку та користувачів задля задоволення їх потреб та надання функціональності у найкоротші строки за ціну, котра є дешевшою ніж у продуктів-замінників.
3	Зворотній зв'язок від користувачів	Можливість отримання необхідної інформації для вдосконалення продукту	Наявність вхідних даних та реакція на них з боку команди розробників задля задоволення потреб та бажань кінцевих користувачів системи кешування даних.
4	Грошова винагорода за рекламу	При достатньому попиту на систему кешування даних можлива комерціалізація продукту на основі реклами задля отримання грошової винагороди для подальшого розвитку продукту та оплати заробітної плати працівникам	Точкова комерціалізація продукту; Введення реклами; Ведення додаткових коштів у проект задля його подальшого розвитку.

Стартап-проект можна впроваджувати на ринок.

Таблиця 4.9. Ступеневий аналіз конкуренції на ринку

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	Тип конкуренції: монополістична	Товар від кожної компанії на ринку, являється недосконалим замінником товару, реалізованого іншими фірмами; На ринку є умови для входу та виходу; Ціна корелює між суперниками;	Розробка продукту з характеристиками, які покривають сфери вживання що не покривають інші товари-замінники; Кореляція цін у відповідності до товарів замінників; Різні типи ліцензій.
2	Рівень конкурентної боротьби: світовий	Всі продукти замінники розроблялись інтернаціональними командами з різних куточків світу, продукти не належать до певної держави, а належать команді розробників	Вихід на ринок збуту продукту з необхідною функціональністю; Налагодження маркетингу на основних Інтернет ресурсах задля охоплення великої кількості потенційних користувачів; Надання бета-версій продукту.
3	Галузева ознака: внутрішньогалузева	Даний тип продукту може використовуватися тільки у сфері розробки ІТ додатків \ продуктів	Надання зручного, інтуїтивно зрозумілого інтерфейсу; Підтримка всім відомих методів взаємодії з середовищем розробки; Наявність документації та он-лайн підтримки.

Закінчення таблиці 4.9.

4	Конкуренція за видами товарів: товарно-видова	Дана конкуренція – конкуренція між товарами одного виду.	Впровадження функціональності яка відсутня у товарів-замінників; Спрощення інтерфейсів; Надання підтримки.
5	Характер конкурентних переваг: цінова та не цінова	Цінові переваги – точкова комерціалізація; Не цінова – надання функціональності, що відсутня у товарах-замінниках.	Надання платних ліцензій лише на критично важливу функціональність для клієнта з певним строком підтримки, що зазначена у відповідній ліцензії; Впровадження унікальної функціональності.
6	За інтенсивністю: марочна	Наявність унікального знаку що відрізняє даний продукт від продуктів-замінників	Впровадження власної назви та власного знаку.

Таблиця 4.10. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Аналогічні системи на ринку	Нові системи на ринку	Розробники бібліотек, фреймворків та алгоритмів, які можна використовувати у розробках.	Можливість гнучкого впровадження системи.	Аналогічні системи
Висновки	Прямі конкуренти намагаються сконцентруватися на інших напрямках свої продуктів.	На даний момент їх додаток який може дати нам конкуренцію не дороблений і не впроваджений.	Постачальники диктують умови збереження даних, які захищають приватність. Також постачальники не дають змогу зловживати нормами.	Клієнти можуть диктувати умови на ринку через повідомлення на форумах бо в полі відгуків в точках продажу додатку.	Можливість введення стандартизації роботи алгоритмів розпізнавання рухомих об'єктів.

Таблиця 4.11. Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1	Прагматичність	Через запуск стартапу система буде не дуже складно з точки зору архітектури перший час. Через певний період із додаванням функціоналу та оптимізації алгоритмів роботи програмний код буде все складнішим. Такий етап наступить не раніше одного року постійної роботи над проектом.
2	Зручність	Оскільки стартап розробляється на багатьох платформах з різною шириною екранів, то зручність використання системи на різних пристроях буде відігравати не малу роль у спроможності конкурувати з іншими гравцями ринку.
3	Швидкість роботи	Швидкість роботи відіграє велику роль для користувачів, оскільки вони не будуть готові чекати декілька хвилин на виведення результату роботи додатку.
4	Оптимізація	Якщо додаток буде дуже часто видавати помилки при роботі, то користувачі не будуть вважати додаток надійним.
5	Налаштування під користувача	Різні люди мають різні звички, які вони використовують, наприклад, якщо є люди, які люблять працювати за додатком де є темні кольори, а є такі люди, які люблять світлі кольори. Можливість редагувати зовнішній вигляд додатку надає значну перевагу серед конкурентів.
6	Приватність	В останні роки приватність людей та інформація щодо них все частіше зловживається шахраями або великими корпораціями, які потребують погодження з умовами доступу до приватної інформації та її обробки.
7	Технічна підтримка	Якщо технічна підтримка компанії буде працювати своєчасно та швидко, то це допоможе зберегти репутацію компанії на відміну від конкурентів, де їй не приділяють увагу.

Надалі проводиться аналіз пропозиції – визначаються загальні риси конкуренції на ринку (табл. 4.9): визначаються тип можливої майбутньої конкуренції та її інтенсивність, рівень конкурентоспроможності за рівнем

конкурентної боротьби, видами товарів і галузевою ознакою.

Проаналізувавши можливості роботи на ринку з огляду на конкурентну ситуацію можна зробити висновок: оскільки кожний з існуючих продуктів не впливає у великій мірі на поточну ситуацію на ринку в цілому, кожний з існуючих продуктів має свою специфічну сферу використання та свої позитивні та негативні сторони щодо рішення певних типів задач, то робота та вихід на даний ринок є можливою і реалізованою задачею. Для виходу на ринок продукт повинен мати функціонал що відсутній у продуктів-аналогів, повинен задовольняти потреби користувачів, мати необхідний та достатній функціонал з конфігурування, підтримку зі сторони розробників та можливість розробки спеціального функціоналу за відповідною ліцензією.

Висновок: проект може бути впроваджений на ринку з огляду на конкурентну ситуацію.

Таблиця 4.12. Порівняльний аналіз сильних та слабких сторін системи
обробки зображень

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з запропонованим						
			-3	-2	-1	0	+1	+2	+3
1	Прагматичність	15	Наш			K1		K2	
2	Зручність	18		K1	Наш		K2		
3	Швидкість роботи	20	K1			K2			Наш
4	Оптимізація	15	K2	Наш	K1				
5	Налаштування під користувача	18			K2	Наш	K1		
6	Приватність	20		K1	K2				Наш
7	Технічна підтримка	17	K2	K1			Наш		

Інтелектуальна система обробки зображень вище завдяки ціновому фактору та новизні запропонованого рішення.

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (Strength, Weak, Opportunities, Troubles) (табл. 4.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін[15].

Проведений SWOT-аналіз показав, що стартап-проект доцільно реалізовувати.

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок. Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів.

Таблиця 4.13. SWOT аналіз стартап-проекту

<p>Сильні сторони (S):</p> <p>прагматичність системи через її легкість роботи; простота у використанні; зменшення затрат у редагуванні; наявність відкритого вихідного коду; збереження приватності інформації користувача.</p>	<p>Слабкі сторони (W):</p> <p>неоптимізованість алгоритму; швидкість роботи системи; потрібне з'єднання з інтернетом</p>
<p>Можливості (O):</p> <p>Зворотній зв'язок з клієнтурою компанії для спроможності розвивати проект в інші напрямки.</p>	<p>Загрози (T):</p> <p>Автономна робота алгоритмів на електронному пристрої; Складність роботи алгоритму при невиявлених випадках використання додатку.</p>

Таблиця 4.14. Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Безкоштовне надання певного функціоналу у користування споживачам на обмежений термін	Головний ресурс – люди, даний ресурс - наявний	2-3 місяці
2	Реклама	Залучення власних коштів для реклами товару	1-2 місяці
3	Написання статей та опис товару на відомих ресурсах	Головний ресурс – час, даний ресурс - наявний	2-3 тижні
4	Презентація товару на хакатонах й інших ІТ заходах	Ресурс – час та гроші для участі, наявні	1-3 місяці

Є можливість ринкової комерціалізації проекту, наявний попит та рентабельність роботи на ринку.

Є перспективи впровадження з огляду на потенційні групи клієнтів, бар'єри входження, стан конкуренції, конкурентоспроможність проекту. Доцільною є подальша імплементація проекту.

4.4 Розроблення ринкової стратегії проекту

Таблиця 4.15. Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Орендарі	Потребують	Попит є	Присутня	Помірно
2	Продавці	Потребують	Попит є, проте нижчий ніж у орендарів	Присутня	Помірно
3	Орендарі	Потребують	Попит є	Присутня	Помірно
4	Покупці	Потребують	Попит є, проте нижчий ніж у орендарів	Присутня	Помірно

Які цільові групи обрано: оскільки різниця між цільовими групами зовсім незначна, а також враховуючи той факт, що компанія має бажання почати продажі (а відповідно і отримання прибутку) якомога швидше, то доцільно враховувати усі цільові групи, тобто використовувати масовий маркетинг, пропонуючи стандартизовану програму.

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку.

Таблиця 4.16. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Надання функціональності що відсутня у товарів-замінників, підтримка клієнтів.	Проведення реклами, освітлення унікальної функціональності через інтернет ресурси та інші канали, контакт напряду з споживачами; формування лояльності і прихильності споживачів.	Зниження ступеню заміненості товару; прихильність клієнтів; відмітні властивості товару; відмітні характеристики товару.	Стратегія диференціації (допускається стратегія спеціалізації).

Таблиця 4.17. Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, які?	Стратегія конкурентної поведінки
Ні, оскільки є товари-замінники, але дані товари замінники не мають деякого необхідного функціоналу	Так, ціль компанії знайти нових споживачів та, частково, забрати існуючих у конкурентів задля задоволення потреб останніх	Компанія частково копіює характеристики товару конкурента, основна ціль компанії розробка нового унікального функціоналу, з підтримкою основного функціоналу конкурентів	Стратегія виклику лідера

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту, а також в залежності від обраної базової стратегії розвитку та стратегії конкурентної поведінки розробляється стратегія позиціонування, що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку або проект.

Таблиця 4.18. Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
Доступна ціна, простота і зручність використання, універсальність	Стратегія диференціації	Вирішення важливих поставлених задач швидко, легко та зрозуміло навіть без інструкцій. Легкість і простота у використанні. Доступність через ціну та технічні характеристики.	стандарти якості; метрики ПЗ; ASQAS – automated system of quality assessment software.

Результатом є узгоджена система рішень щодо ринкової поведінки стартап-компанії, яка визначатиме напрями роботи стартап-компанії на ринку. Отже, робота стартап-компанії на ринку повинна бути спланована орієнтовано таким чином: за стратегією диференціації виконаний і буде поширюватись товар відмінний за властивостями від своїх аналогів, дотримуючись у конкурентній поведінці стратегії «виклику лідера», тобто випускається один товар для усіх можливих споживачів.

Надалі розроблена трирівнева маркетингова модель товару: уточнюються ідея продукту, його фізичні складові, особливості процесу його надання.

4.5 Розроблення маркетингової програми стартап-проекту

Першим кроком є формування маркетингової концепції товару, який отримає споживач.

Таблиця 4.19. Визначення ключових переваг концепції потенційного товару

Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
Інтелектуальна система обробки зображень.	Отримання з зображень об'єктів їх тривимірну модель та лінійні розміри.	Розрахункові показники, точність та достовірність яких можна оцінювати; простота, кількість вхідних параметрів; самостійність програмної системи.

Таблиця 4.20. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
Товар за задумом	Створення користувацьких плагінів; інтеграція з популярними CAD системами; локальна обробка зображень; хмарна обробка зображень; створення та редагування тривимірних моделей по зображенню; аналітичний блок оцінки параметрів об'єкта.
Реалізований товар	Підтримка в режимі роботи дизайнера та будівельника; перегляд списку робіт; хмарна обробка зображень; перегляд статусу робіт; створення тривимірних моделей по зображенню.

За рахунок чого потенційний товар буде захищено від копіювання: від копіювання потенційний товар захистити не складає проблеми. Розроблена математична модель підбору пропозицій, на якій базується програмна система, публікувалась лише у загальних рисах, а без математичної моделі цей ПП лише набір рядків коду.

Визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар, яке передбачає аналіз ціни на товари-аналоги, а також аналіз рівня доходів цільової групи споживачів.

Таблиця 4.21. Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
10000 грн	10000 грн	40000 грн	4000-30000 грн

Таблиця 4.22. Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Бажання отримати більше за менші гроші	Залучення клієнтської бази та продаж	Нульовий рівень: тільки виробник	Вертикальна маркетингова система

Таблиця 4.23. Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Бажання отримати більше за менші гроші	Будь-які, але бажано з великою кількістю візуального контенту	Низька ціна Широкий вибір функціоналу Легкий і простий у використанні продукт	Донести до користувача суть продукту, його якість, та залучити якомога більше зацікавлених клієнтів	Донести до користувача суть продукту, його якість, та залучити якомога більше зацікавлених клієнтів

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 4.21): проводити збут власними силами або залучати сторонніх посередників, вибір та обґрунтування оптимальної глибини каналу збуту, вибір та обґрунтування виду посередників.

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.22).

Як результат було створено ринкову (маркетингову) програму, що включає в себе визначення ключових переваг концепції потенційного товару, опис моделі товару, визначення меж встановлення ціни, формування системи збуту та концепцію маркетингових комунікацій.

Висновки по розділу

В четвертому розділі описано стратегії та підходи з розроблення стартап-проекту, визначено наявність попиту, динаміку та рентабельність роботи ринку, як висновок було вказано що існує можливість ринкової комерціалізації проекту. Розглянувши потенційні групи клієнтів, бар'єри входження, стан конкуренції та конкурентоспроможність проекту було встановлено що проект є перспективним. Розглянуто та вибрано альтернативу впровадження стартап-проекту та доведено доцільність подальшої імплементації проекту.

Отже, ринкова (маркетингова) програма орієнтовано має бути побудована таким чином:

- розробка продукту;
- вибір сегменту ринку та пошук клієнтів;
- стратегія розвитку – стратегія розподіленості, тобто формування конкурентоспроможності досягається шляхом надання споживачу товару, якого той потребує. На основі детального вивчення середовища споживання розробляється одна або декілька особливих характеристик власного товару;

– стратегія конкурентної поведінки – стратегія виклику лідера, тобто на споживчому ринку націлюватись на всіх можливих споживачів, у тому числі клієнтів фірм-конкурентів. Така стратегія будується за принципом «йти слідом» за лідером ринку. За наступні цілі ставиться можливість обійти лідерів цільового сегменту.

Стан та динаміка ринкового середовища на сьогоднішній день і ще багато років є і будуть залишатись сприятливими для впровадження розробленої системи, а також для її необхідності.

Конкурентні переваги створеного продукту очевидні. На вітчизняному ринку аналогів майже не існує, а існуючі – вкрай низької якості. На міжнародному ринку конкуренція наявна та буде рости, якщо не підтримувати та не розвивати свій продукт.

Також, після проведення аналізів можливого цільового сегменту (споживачів), потреб споживачів та можливого попиту, динаміки ринку та рентабельності роботи на ринку, можна однозначно зробити висновок, що створений проект доцільний до комерціалізації.

Перспективи впровадження з огляду на потенційні групи клієнтів, бар'єри входження, стан конкуренції та конкурентоспроможності проекту – прямі, і тільки доводять можливість впровадження, та не марну розробку створеного продукту.

ВИСНОВКИ

Важливою характеристикою роботизованих пристроїв є ефективне використання допоміжного обладнання, що надасть змогу оптимізувати кількість датчиків на роботизованих пристроях, тобто використання камери може замінити велику кількість датчиків. Визначено основні проблеми у реалізації систем визначення характеристик з об'єктів на зображенні. Для подолання цієї використано нейромережі, як сучасний спосіб обробки графічної інформації з камери. Сформовано функціонал системи, визначені бізнес-процеси та ролі.

Досліджено архітектуру та технології розробки для побудови роботизованої системи по розпізнаванню характеристик об'єктів. У результаті обрано використовувати: Visual Studio Code; Node.js, Express, JavaScript – для програмної реалізації та інтерфейсу користувача, мову Python та фреймворк TensorFlow – для розробки нейронної мережі. Використання цих сучасних технологій дозволяє досягти високих показників ефективності системи за мінімальну вартість.

Створено як мікросервіс підсистему для отримання тривимірних моделей з зображення та оцінка їх параметрів.

Використання мікросервісної архітектури дає можливість легко масштабувати систему під впливом обчислювального навантаження на систему, що значно спрощує задачу удосконалення та адаптування системи при збільшенні кількості роботизованих пристроїв.

Для оцінки ефективності розробленої системи виконано моделювання роботи кількох типів нейромереж для генерації тривимірних моделей воксельним та сітковим методом для оцінки параметрів об'єкта.

Експериментально доведено, що похибка в отриманих нейромережею тривимірних об'єктах є не значною у порівнянні з складними алгоритмами які вимагають великої кількості зображень для створення моделі, при чому

результатом роботи може бути не суцільна модель, а хмара точок яка вимагає ще додаткової обробки.

Результатом магістерської роботи стала робототехнічна система визначення характеристик об'єктів по зображеннях. Система оснащена зручним користувацьким інтерфейсом та задовольняє усі функціональні вимоги системи.

ПЕРЕЛІК ПОСИЛАНЬ

1. Photogrammetry [Електронний ресурс] // Режим доступу: <https://en.wikipedia.org/wiki/Photogrammetry>.
2. ASPRS online Archived May 20, 2015, at the Wayback Machine.
3. Sužiedelytė-Visockienė J, Bagdžiūnaitė R, Malys N, Maliene V (2015). «Close-range photogrammetry enables documentation of environment-induced deformation of architectural heritage» *Environmental Engineering and Management Journal*. 14 (6): 1371–1381. doi:10.30638/eemj.2015.149.
4. Ina Jarve, Natalja Liba. The Effect of Various Principles of External Orientation on the Overall Triangulation Accuracy. *TECHNOLOGIJOS MOKSLAI*. Estonia. #86, 2010, pp. 59-64.
5. 2019 Best Photogrammetry Software [Електронний ресурс] // Режим доступу: <https://all3dp.com/1/best-photogrammetry-software/>.
6. S. Ullman (1979). «The interpretation of structure from motion» (PDF). *Proceedings of the Royal Society of London*. 203 (1153): 405 – 426.
7. openMVG documentation [Електронний ресурс] // Режим доступу: <https://openmvg.readthedocs.io/en/latest/>.
8. Agisoft Metashape [Електронний ресурс] // Режим доступу: <https://en.wikipedia.org/wiki/Metashape>.
9. RealityCapture [Електронний ресурс] // Режим доступу: <https://en.wikipedia.org/wiki/RealityCapture>.
10. PhotoModeler [Електронний ресурс] // Режим доступу: <https://en.wikipedia.org/wiki/PhotoModeler>.
11. 3DF Zephyr [Електронний ресурс] // Режим доступу: https://en.wikipedia.org/wiki/3DF_Zephyr.
12. Pix4D [Електронний ресурс] // Режим доступу: <https://en.wikipedia.org/wiki/Pix4D>.
13. What is Photogrammetry? [Електронний ресурс] // Режим доступу: <http://www.photogrammetry.com/index.htm>.

14. 3D data acquisition and reconstruction [Електронний ресурс] // Режим доступу: https://en.wikipedia.org/wiki/3D_data_acquisition_and_object_reconstruction.
15. Розроблення стартап-проекту [Електронний ресурс] : Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ : НТУУ «КПІ», 2016. – 28 с.
16. Ren, Xiaoyuan, Libing Jiang, Xiaonan Tang, and Weichun Liu. «3D Wireframe Modeling and Viewpoint Estimation for Multi-Class Objects Combining Deep Neural Network and Deformable Model Matching.» *Applied Sciences* 9, no. 10 (2019): 1975.
17. Choy, Christopher B., Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. «3d-r2n2: A unified approach for single and multi-view 3d object reconstruction». In *European conference on computer vision*, pp. 628-644. Springer, Cham, 2016.
18. Wu, Jiajun, Tianfan Xue, Joseph J. Lim, Yuandong Tian, Joshua B. Tenenbaum, Antonio Torralba, and William T. Freeman. «3D interpreter networks for viewer-centered wireframe modeling.» *International Journal of Computer Vision* 126, no. 9 (2018): 1009-1026.
19. Zhang, Cha, and Tsuhan Chen. «Efficient feature extraction for 2D/3D objects in mesh representation.» In *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, vol. 3, pp. 935-938. IEEE, 2001.
20. Mousavian, Arsalan, Dragomir Anguelov, John Flynn, and Jana Kosecka. "3d bounding box estimation using deep learning and geometry." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7074-7082. 2017.

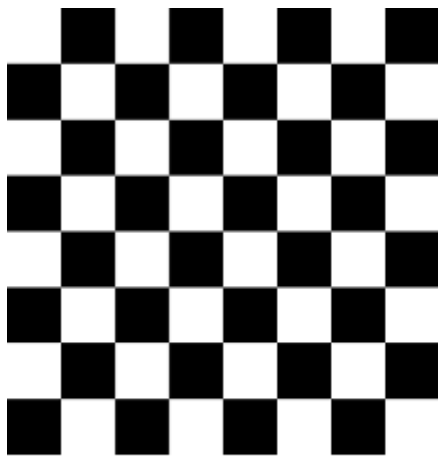
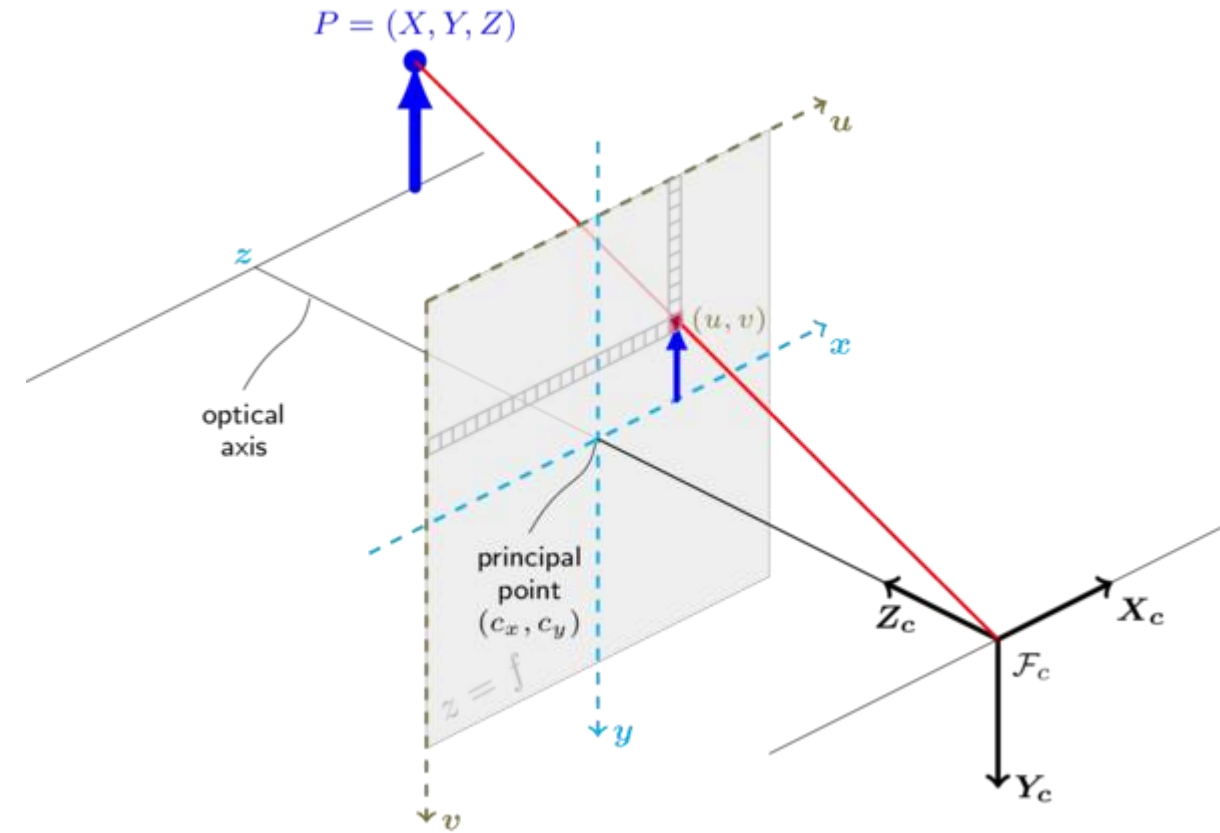
ДОДАТКИ

ДОДАТОК А

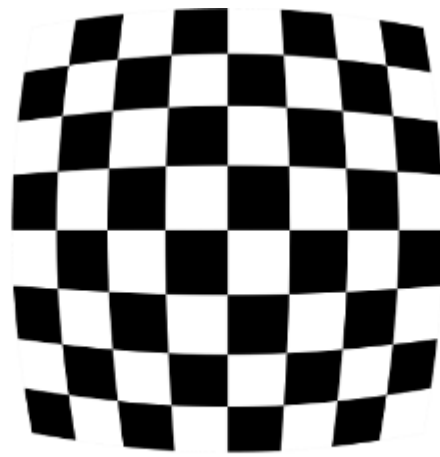
Звіт про перевірку на співпадіння

ДОДАТОК Б
Графічні матеріали

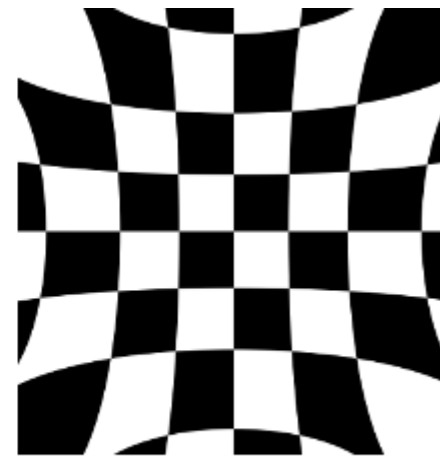
Калібрування камери



No distortion



Positive radial distortion
(Barrel distortion)



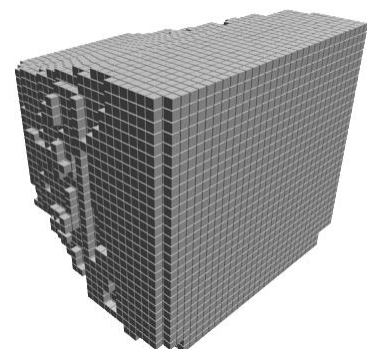
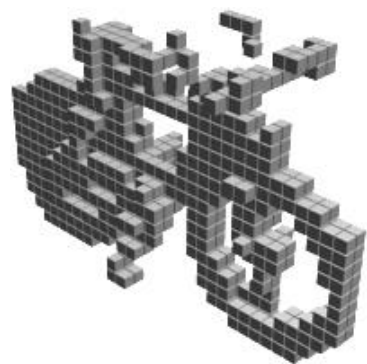
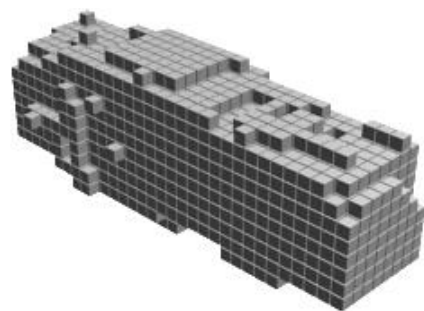
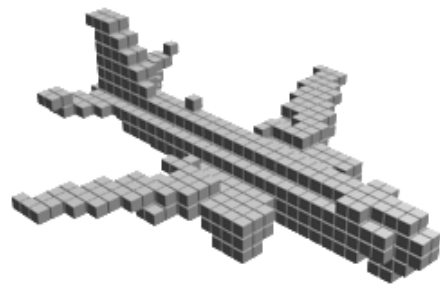
Negative radial distortion
(Pincushion distortion)

Демонстраційний плакат №1
до магістерської дисертації на тему
«Робототехнічна система визначення характеристик об'єктів по
зображеннях»

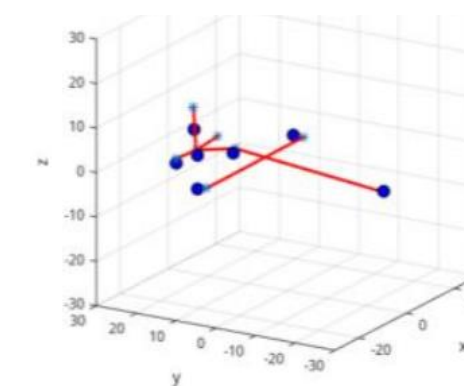
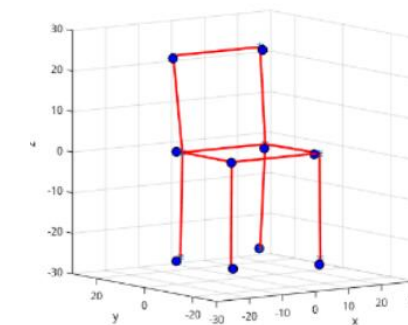
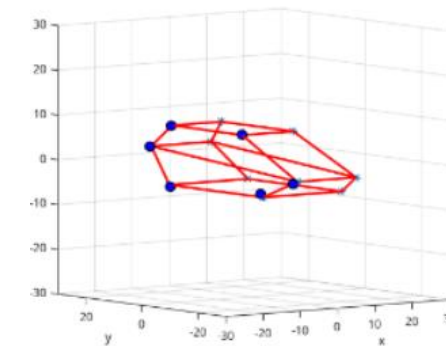
Розробив: Гаврилюк Ю.О.
Прийняв: к.т.н., доцент Корнага Я.І.

Підходи в генеруванні 3D-моделей

Воксельний



Каркасный

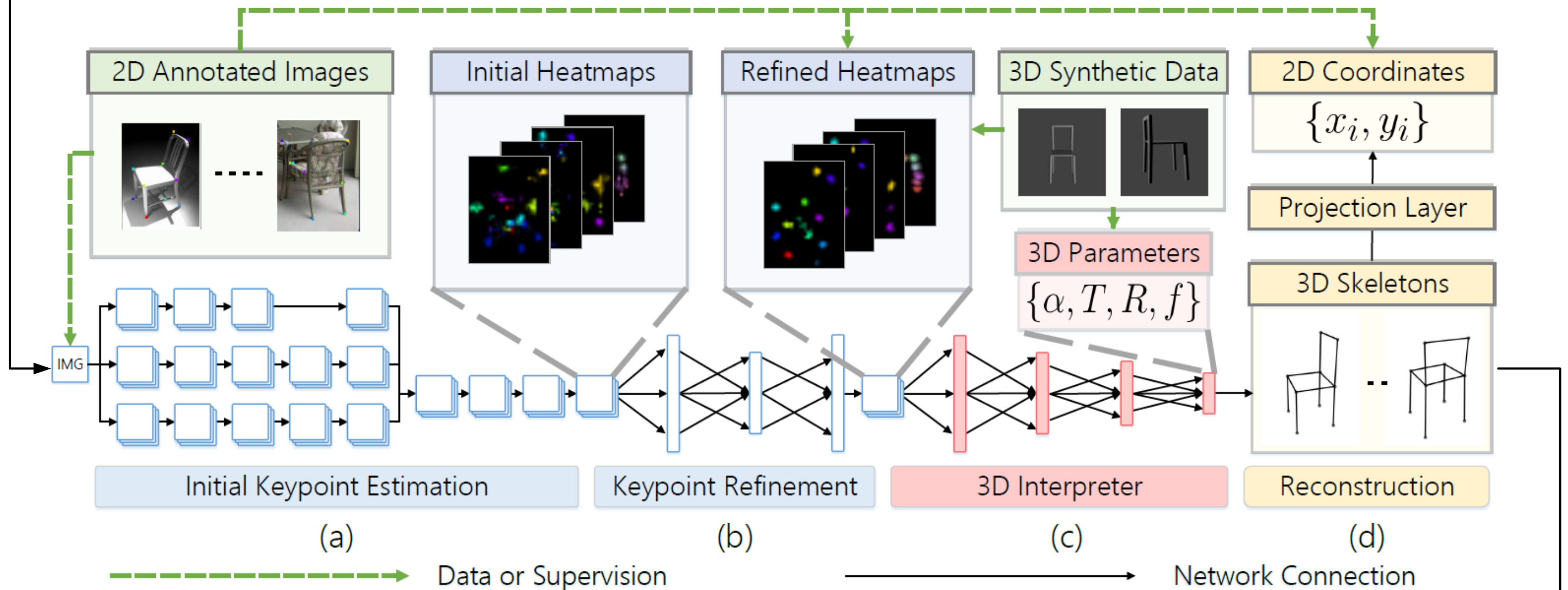


Демонстраційний плакат №2
до магістерської дисертації на тему
«Робототехнічна система визначення характеристик об'єктів по
зображеннях»

Розробив: Гаврилюк Ю.О.
Прийняв: к.т.н., доцент Корнага Я.І.

Архітектура системи

Нейромережа по
виділенню об'єктів

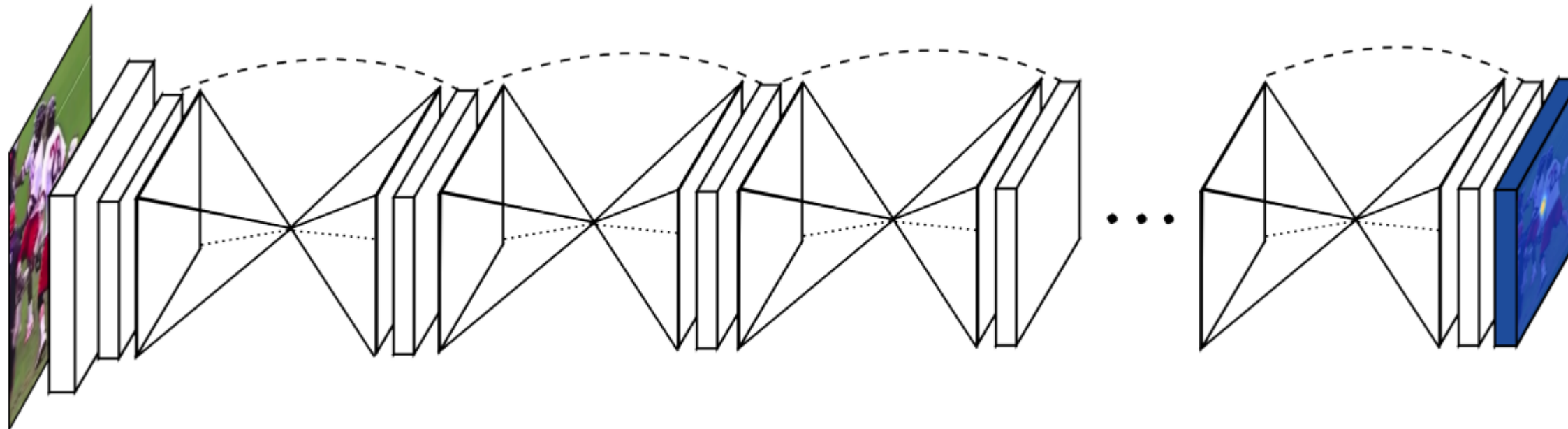


Модуль оцінки
характеристик

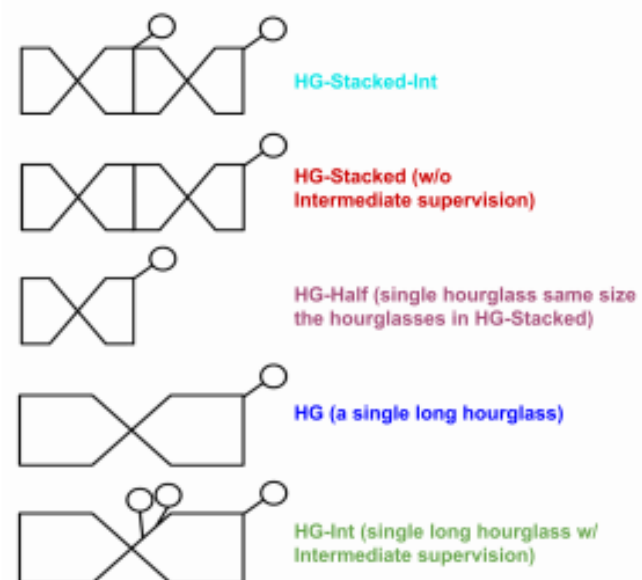
Демонстраційний плакат №3
до магістерської дисертації на тему
«Робототехнічна система визначення характеристик об'єктів по
зображеннях»

Розробив: Гаврилюк Ю.О.
Прийняв: к.т.н., доцент Корнага Я.І.

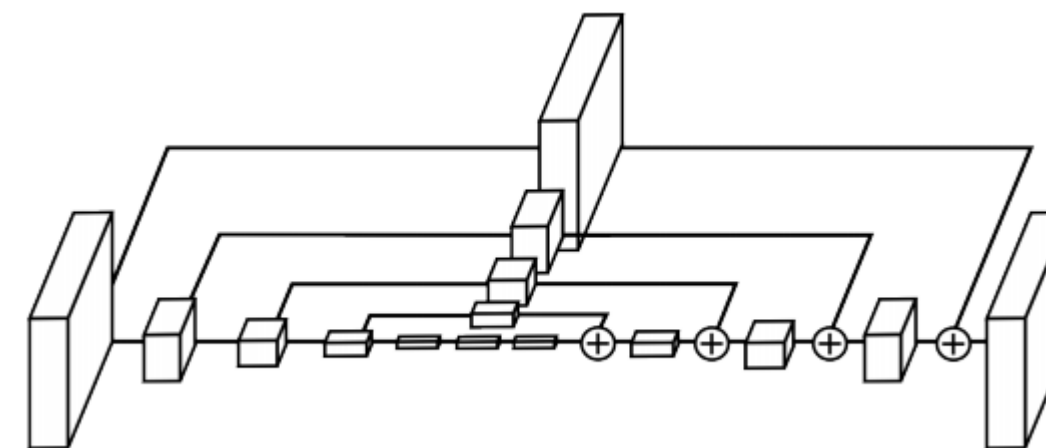
Архітектура нейромережі Hourglass



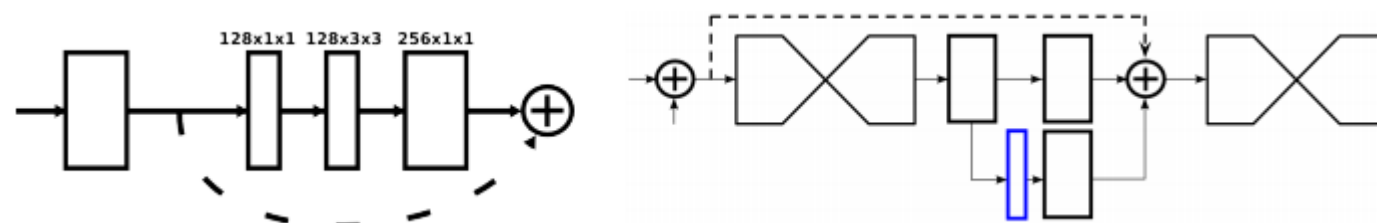
Можливі комбінації



Одиничний компонент



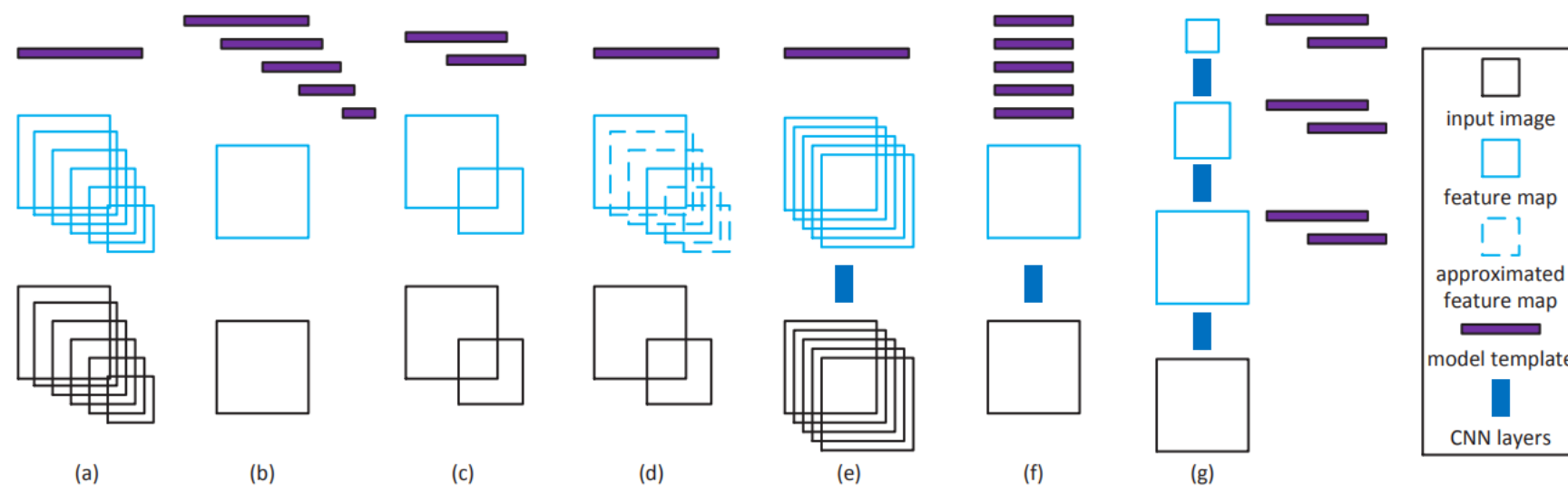
Додаткові складові



Демонстраційний плакат №4
до магістерської дисертації на тему
«Робототехнічна система визначення характеристик об'єктів по
зображеннях»

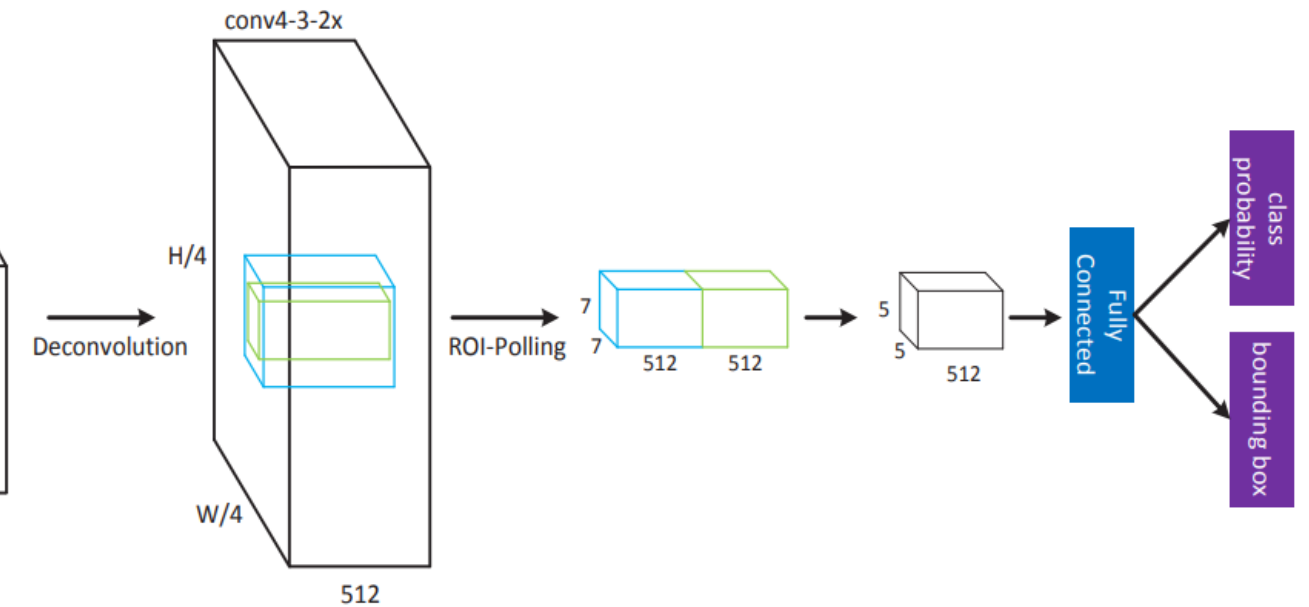
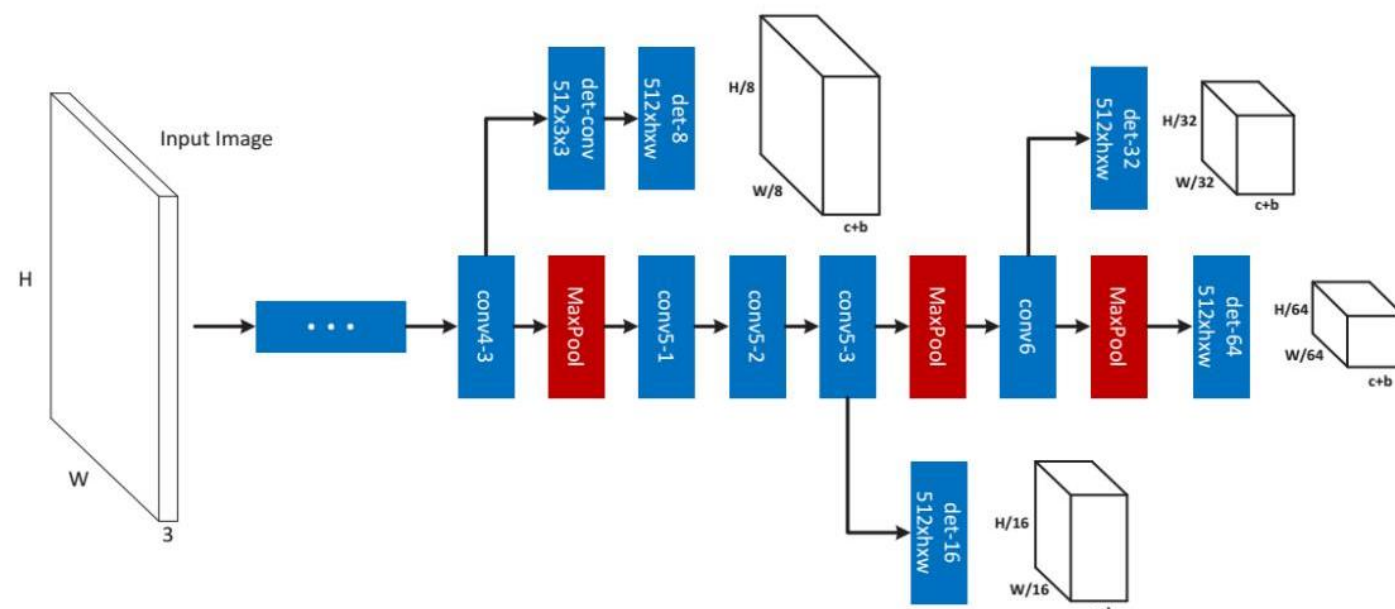
Розробив: Гаврилюк Ю.О.
Прийняв: к.т.н., доцент Корнага Я.І.

Архітектура нейромережі MS-CNN



Вихід з мережі

Вхід мережі

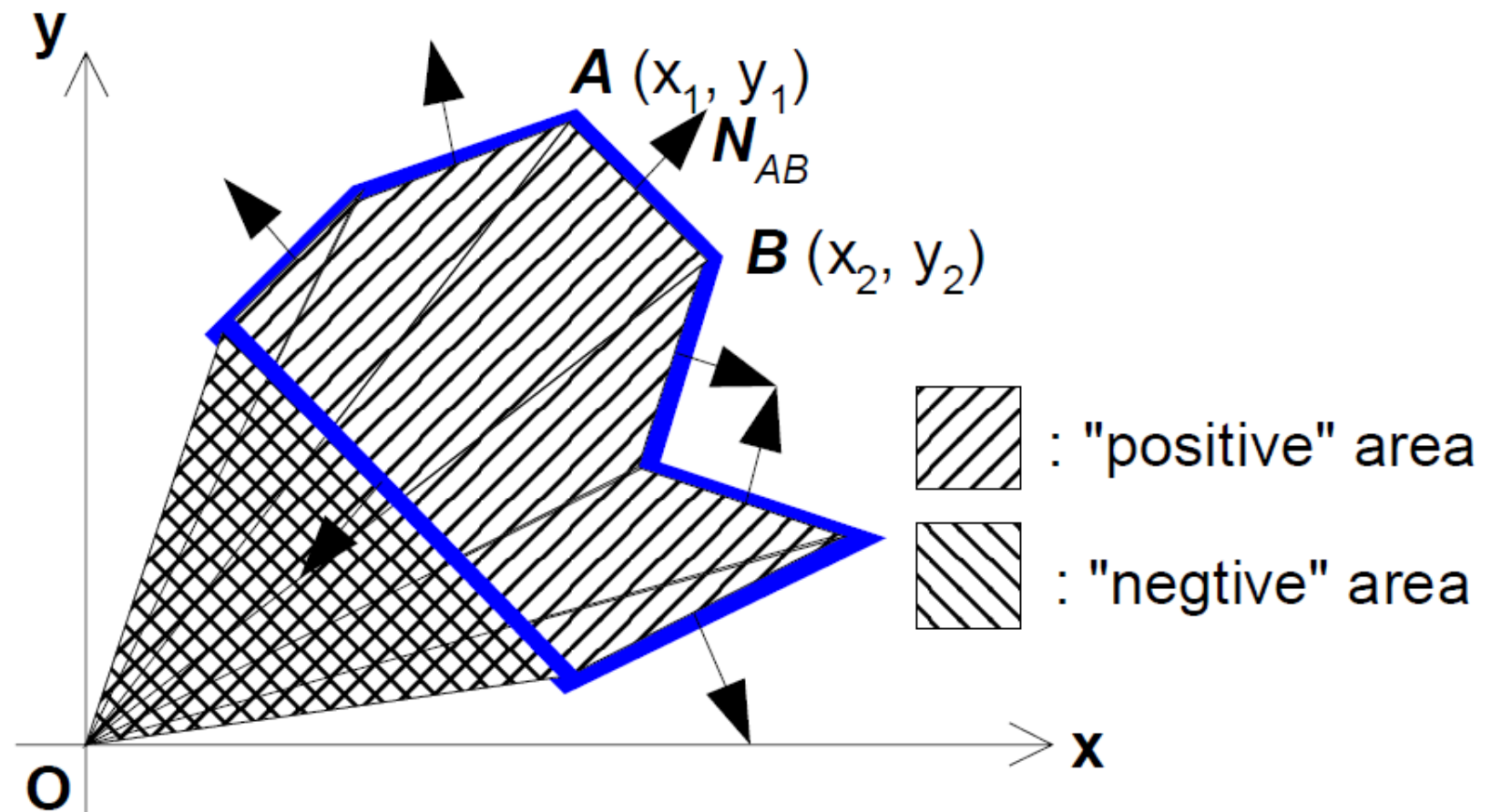


Демонстраційний плакат №5
до магістерської дисертації на тему
«Робототехнічна система визначення характеристик об'єктів по
зображеннях»

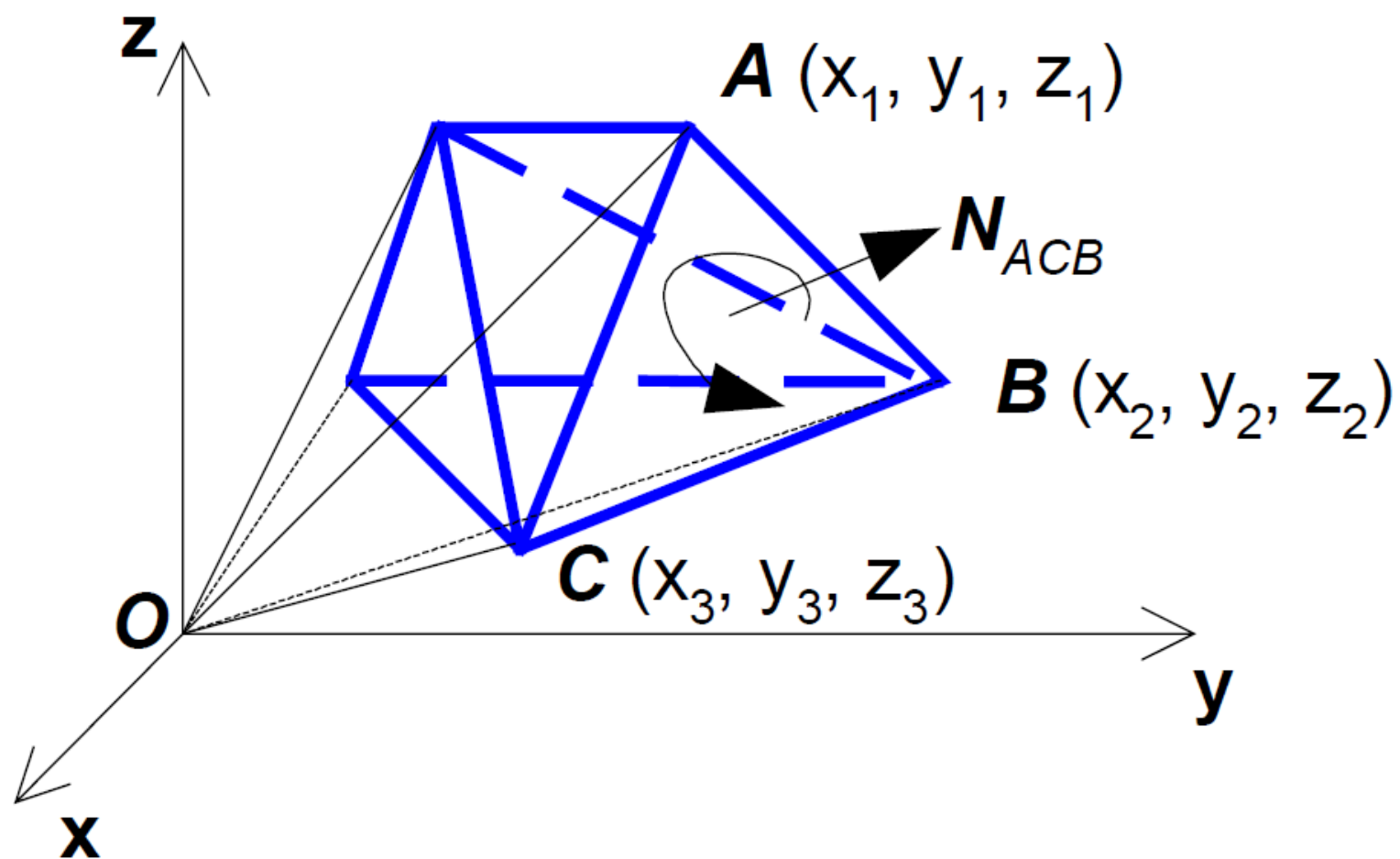
Розробив: Гаврилюк Ю.О.
Прийняв: к.т.н., доцент Корнага Я.І.

Отримання характеристик об'єктів

Площа поверхні



Об'єм



Демонстраційний плакат №6
до магістерської дисертації на тему
«Робототехнічна система визначення характеристик об'єктів по
зображеннях»

Розробив: Гаврилюк Ю.О.
Прийняв: к.т.н., доцент Корнага Я.І.